

Generative AI-Enhanced Real-Time Anomaly Detection in Integrated Energy Systems

Sobhan Badakhshan¹, Graduate Student Member, IEEE, and Jie Zhang², Senior Member, IEEE

Abstract—The integration of Integrated Energy Systems (IES) with main power grids is vital for enhancing the efficiency, security, and resilience of modern energy systems. However, this increased interconnectivity among energy sources, smart grids, and digital monitoring systems also exposes IES to significant cyber threats. To mitigate these risks, we have developed a framework for real-time, AI-assisted monitoring and anomaly detection, leveraging generative AI models to monitor inter-connected IES within the main grid. This paper presents a Generative Adversarial Networks (GAN)-based prediction and anomaly detection system, specifically leveraging the Wasserstein GAN with Gradient Penalty (WGAN-GP) with Long Short-Term Memory (LSTM), for secure monitoring of grid-connected IES. Our approach combines the generator’s predictive capabilities with the discriminator’s scoring mechanism to enhance anomaly detection and overall model accuracy. This allows the system to forecast the control system’s future response and detect anomalies before they manifest in real-time data. The effectiveness of this framework is assessed using an interconnected IES to the IEEE 118-bus test network. The pre-trained model is subjected to diverse attack scenarios, and experimental results consistently demonstrate its capability to identify the probability of anomalies within the IES efficiently.

Index Terms—Generative AI, cybersecurity, integrated energy systems, generative adversarial networks, anomaly detection.

I. INTRODUCTION

INTEGRATED Energy Systems (IES) are vital in optimizing resource utilization and promoting sustainability in modern energy management. These systems interconnect different energy sources, such as renewable energies, batteries, clean fuels, and energy storage, to efficiently serve the needs of both local and regional consumers [1]. IES combines multiple energy sources and technologies, either tightly or loosely coupled, to deliver electricity, heat, transportation, and other energy services. It enhances the overall efficiency, reliability,

and sustainability of the energy supply. [2]. Moreover, it contributes to enhancing grid resilience and promoting community involvement as a crucial step towards achieving 100% renewable energy [3].

Numerous research has investigated the optimal operation and design of IES [4]. For instance, the decentralized operation of an IES in Great Britain has been used to simulate a decarbonized energy system with a multi-vector energy strategy. The simulation shows improvements in operational flexibility, better use of renewable resources, and a reduced need for large investments in expanding the electricity transmission network [5]. A distributed solar-biogas residential IES is designed and optimized to supply thermal, electrical, and gas loads in remote locations [6]. Integrating IES with the main grid enhances energy efficiency, cost-effectiveness, reliability, sustainability, and resilience [7]. Its diverse energy mix offers operational flexibility and supports net-zero goals [8], while also improving community resilience after extreme events [9]. However, integrating IES with the main grid introduces cybersecurity challenges, particularly for real-time monitoring and dynamic anomaly detection. Cyberattacks on IES can destabilize the grid, cause financial losses, and deceive operators. In competitive markets, attackers may target other IES for profit or influence. Therefore, detecting and mitigating threats at the grid-IES interface is essential.

Numerous studies have explored anomaly detection approaches for real-time cybersecurity monitoring of both power systems and IES. For instance, [10] examines targeted cyber-attacks on different IES components, including heat load redistribution attacks, revealing latent vulnerabilities and security challenges. Machine learning techniques, such as those in [11], utilize conditional variational autoencoders with attention mechanisms for the classification and detection of anomaly patterns in IES. Additionally, [12] presents data-driven approaches for detecting anomalies and analyzing vulnerability dynamics in large-scale IES. Furthermore, a time-frequency feature prediction method for anomaly detection in cyber-physical IES is outlined in [13]. The time series analysis method involves examining data patterns over time to detect anomalies, with k-nearest neighbor analysis [14], [15]. A sophisticated graph-structure-based framework for pattern recognition and spectral clustering is discussed for hybrid energy systems in [12] and [16] to enhance the security of systems. Probability-based models, like Bayesian networks, could be applied to predict cybersecurity threat likelihoods in a network [17]. The deployment of artificial neural networks,

Received 11 January 2025; revised 28 June 2025 and 28 September 2025; accepted 4 November 2025. Date of publication 7 November 2025; date of current version 23 February 2026. This work was supported by the U.S. Department of Energy (DOE) through the Idaho National Laboratory (INL) Directed Research and Development (LDRD) Program under DOE Idaho Operations Office Contract DE-AC07-05ID14517. Paper no. TSG-00055-2025. (Corresponding author: Jie Zhang.)

Sobhan Badakhshan is with the Department of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX 75080 USA. Jie Zhang is with the Department of Mechanical Engineering and the Department of Electrical and Computer Engineering (Affiliated), The University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: jiezhang@utdallas.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TSG.2025.3630562>.

Digital Object Identifier 10.1109/TSG.2025.3630562

1949-3053 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on March 25, 2026 at 18:26:23 UTC from IEEE Xplore. Restrictions apply.

especially Convolutional Neural Networks (CNNs), is valuable for learning complex patterns and detecting anomalies, as highlighted by [18]. For example, [19] introduces a deep reinforcement learning-based feature selection model for network intrusion detection. CNN-based methods use convolutional layers to extract features from labeled time series data (normal and anomalous) and fully connected layers for unsupervised classification, learning to distinguish key patterns. However, in scenarios with new technologies where labeled data for rare anomalies is scarce or unavailable, traditional deep-learning cybersecurity models become impractical. To overcome this, generative methods can be used to generate realistic data, making it possible to address the challenge of limited or costly labeled anomalous data. Within the energy sector, generative AI plays a pivotal role in predictive analytics, smart grid management [20], market forecasting, and integration of renewable energy sources. In the power system dynamic security assessment, [21] introduces a data-driven GAN model to address missing data, enhancing the overall robustness of the assessment process. Generative AI is effective for real-time anomaly detection in IES by learning from unlabeled, heterogeneous time-series data using simulation-based testbeds. GANs, in particular, enable dynamic feature extraction through adversarial training and can detect subtle anomalies by assigning probability scores for unseen patterns that reflect how well new signals match learned normal patterns. However, the application of generative models in anomaly detection and operational security of the energy system is relatively new and underexplored. One key motivation for using a pre-trained GAN-based model is its ability to plug into existing systems without requiring substantial upgrades to the infrastructure.

In this paper, we present a novel approach to real-time anomaly detection in complex energy systems using a Wasserstein GAN with Gradient Penalty (WGAN-GP) combined with a Long Short-Term Memory (LSTM) architecture. The LSTM-enhanced GAN incorporates LSTM layers to model time-series data. The latent vector input to the generator is passed through the LSTM network, which processes the sequences step-by-step, enabling the model to learn temporal dependencies and generate sequential data that closely mimics the time-dependent structure of the real data. The methodology combines predictive modeling with an improved discriminator score from a pre-trained generator and discriminator to detect anomalies accurately. The LSTM-GAN model extracts features from real datasets, adapts to evolving patterns, and improves anomaly detection accuracy. This addresses key challenges in IES, particularly those related to modeling non-stationary behavior, the interaction between thermal and electrical subsystems, and the limited adaptability of conventional detection methods in such multi-domain environments. Unlike traditional machine learning techniques, the proposed framework enables adaptive, unsupervised anomaly detection. By leveraging the synthetic data generation capabilities of LSTM-GAN, they can identify previously unseen anomalies that traditional methods might miss.

To train the model, we develop dynamic models of IES, including renewable energy resources, batteries, thermal units, and main grids across various operating scenarios. This

dataset enables pre-trained discriminators to accurately model complex power grid behaviors, supporting early anomaly detection. The LSTM-GAN model sets a probability score threshold to detect cyberattacks in real time, triggering alarms when exceeded. The proposed model significantly reduces the incidence of false positives and negatives, thereby delivering more accurate anomaly detection.

The rest of the article is organized as follows: Section II reviews the design of the IES and the importance of anomaly detection in its operation. In Section III, we develop an AI-assisted real-time monitoring approach for anomaly detection using the WGAN-GP. Numerical results are presented in Section IV, and conclusions are drawn in Section V.

II. INTEGRATED ENERGY SYSTEMS (IES) DESIGN

IES refers to the concept of combining diverse energy sources, technologies, and infrastructure to optimize the entire energy life cycle, from generation and distribution to utilization. This framework represents an electro-thermal IES without modeling the gas network, with a focus on applications where electro-thermal interactions dominate, such as cogeneration in combined heat and power units, waste-heat utilization for cooling and data center energy management, and renewable-thermal hybrid systems. As illustrated in Fig. 1, in this paper, we have created an IES that seamlessly integrates various energy sources, including renewable options like solar, wind, and battery, along with clean fuels and thermal demand. Dynamic modeling of a grid-connected IES captures the interaction between synchronous generators, inverter-based resources, and the power grid, using equations to represent system dynamics, steady states, and disturbance responses.

The dynamics of synchronous generators, including their governors, Automatic Voltage Regulators (AVRs), and stabilizers, are described by differential equations:

$$\dot{x} = f(x, u), \quad x \in \mathbb{R}^{n_{\text{states}}}, \quad (1)$$

where x represents dynamic states such as rotor angle, angular velocity, and excitation voltage, while u denotes control inputs like governor settings and excitation voltage. These equations model the electromechanical behavior of generators and their controllers. For inverter-based resources such as batteries, solar panels, and wind turbines, the dynamics are represented similarly but account for the power-electronic interfaces. Their state-space equations incorporate the control strategies regulating active and reactive power, as well as voltage and frequency. The power balance at each node in the system is represented as algebraic constraints that ensure the sum of active and reactive power injections and withdrawals balance to zero in steady-state and adjust dynamically:

$$0 = g(x, y), \quad y_1 \in \mathbb{R}^{n_{\text{algebraic}}}, \quad (2)$$

where y_1 includes algebraic variables such as bus voltages and angles, and $g(x, y)$ represents active and reactive power mismatch equations. For transient stability studies, power system simulator PSS®E solves the full set of nonlinear differential-algebraic equations:

$$\dot{x} = f(x, y, u), \quad (3)$$

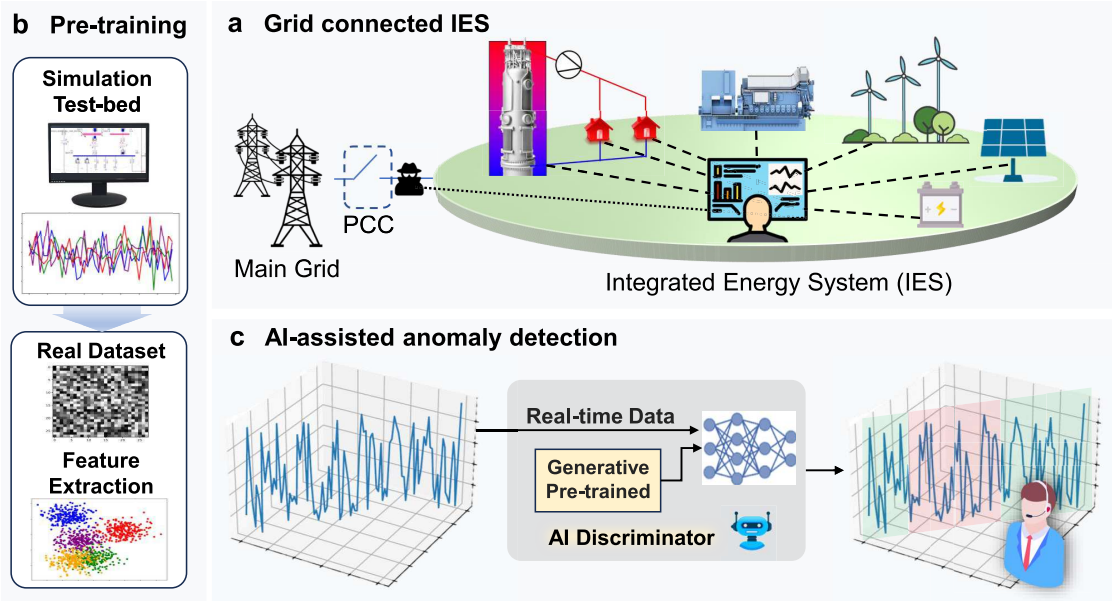


Fig. 1. Schematic illustration of generative AI-assisted anomaly detection in real-time monitoring.

$$0 = g(x, y, u), \quad (4)$$

using numerical integration methods (e.g., trapezoidal rule) over small time steps. This ensures that all nonlinear behaviors are fully preserved during simulation, regardless of the system operating point or disturbance type. Consequently, the time-domain trajectories generated from PSS[®] E reflect the complete nonlinear dynamics of the system and are used to train the AI model. As such, the model is not limited by assumptions of linearization and can capture a wide range of operating behaviors. Although time is inherently included in the numerical integration process, it allows the evolution of states and algebraic variables over time. This approach models the IES's responses to faults, disturbances, and dynamic interactions. The thermal-electrical system dynamics are described, including thermal setpoints, thermal power, and electrical outputs. The system has feedback loops where thermal and electrical powers influence each other. The thermal load changes based on the external setpoint, and its dynamics are modeled in Eq. 7. Thermal power impacts the electrical output, which is controlled by a governor-like mechanism. The rotor speed models the interaction between mechanical and electrical power, considering efficiency losses. We use discretization to approximate the system's behavior in small time intervals for simulation and analysis. The thermal setpoint dynamics are governed by the following differential equation, which models the change in thermal setpoint over time:

$$\frac{dP_{th,set}}{dt} = \frac{1}{\tau_{set}}(P_{th,set,raw} - P_{th,set}), \quad (5)$$

where the raw thermal setpoint, $P_{th,set,raw}$, is determined by the external setpoint and the electrical power:

$$P_{th,set,raw} = \min(P_{th,max}, \max(\text{Setpoint}, k \cdot (P_{el,max} - P_{el}))), \quad (6)$$

with the parameters $P_{th,max}$ and $P_{el,max}$ representing the maximum thermal and electrical power, respectively, and k as a

scaling factor. The thermal power dynamics are described by the following equation, which captures the rate of change of thermal power as it approaches the setpoint:

$$\frac{dP_{th}}{dt} = \frac{1}{\tau}(P_{th,set} - P_{th}), \quad (7)$$

where τ is the thermal power time constant. The electrical output dynamics are governed by the following equation, which models the governor-like behavior of the system:

$$\frac{dP_{el}}{dt} = \frac{1}{\tau_g}(P_{el,th} - P_{el} - K_d \cdot (\omega - \omega_{ref})), \quad (8)$$

where $P_{el,th}$ is the thermal electrical power, given by:

$$P_{el,th} = \max\left(0.0, P_{el,max} - \frac{P_{th,set}}{k}\right), \quad (9)$$

and K_d represents the damping gain while ω is the rotor speed. The rotor speed dynamics are described by the following equation, which models the rotor speed's rate of change in response to mechanical and electrical power:

$$\frac{d\omega}{dt} = \frac{1}{2H}(P_m - P_{el} - D \cdot (\omega - \omega_{ref})), \quad (10)$$

where P_m is the mechanical power, and D is the damping coefficient. The term $P_{el} \approx P_m \cdot \eta$ accounts for efficiency losses. The discretized thermal setpoint update is expressed as:

$$P_{th,set}(t + \Delta t) = P_{th,set}(t) + \Delta t \cdot \frac{P_{th,set,raw} - P_{th,set}(t)}{\tau_{set}}, \quad (11)$$

which approximates the thermal setpoint at the next time step using a finite difference method. The discretized thermal power update is given by:

$$P_{th}(t + \Delta t) = P_{th}(t) + \Delta t \cdot \frac{P_{th,set}(t) - P_{th}(t)}{\tau}, \quad (12)$$

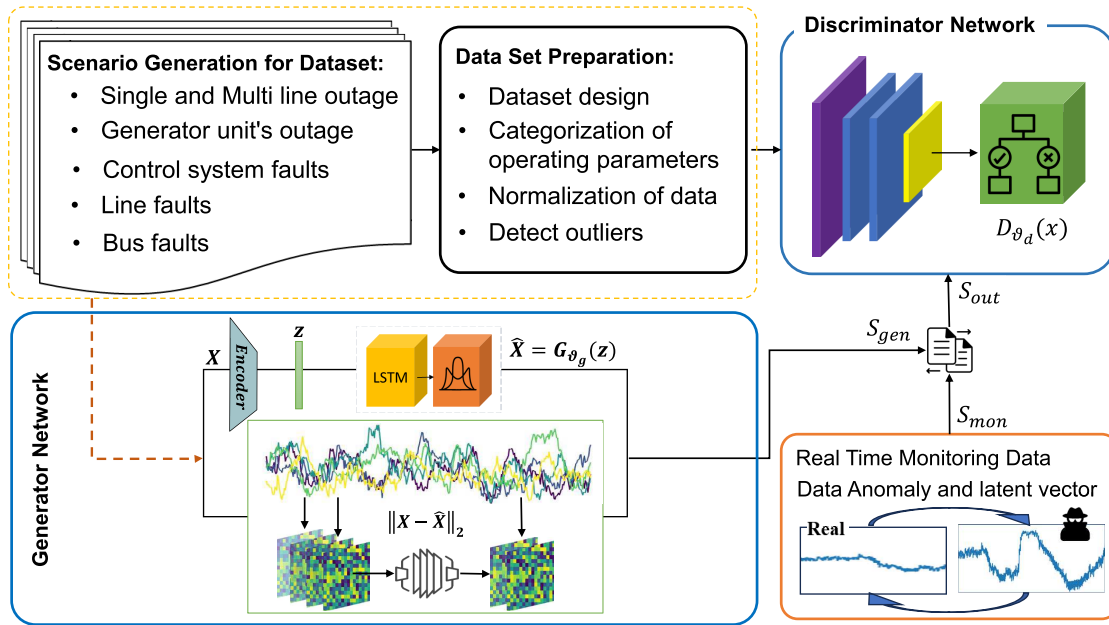


Fig. 2. The GAN-based model detects real-time anomalies by using reconstruction errors and discriminator outputs, leveraging pre-trained models and simulation testbed data.

which models the thermal power change over time using the thermal time constant. The discretized electrical output update is formulated as:

$$P_{el}(t + \Delta t) = P_{el}(t) + \frac{\Delta t}{\tau_g} (P_{el,th}(t) - P_{el}(t) - K_d \cdot (\omega(t) - \omega_{ref})), \quad (13)$$

which approximates the electrical output change at each time step using the electrical time constant and damping gain. Finally, the discretized rotor speed update is:

$$\omega(t + \Delta t) = \omega(t) + \Delta t \cdot \frac{1}{2H} (P_m(t) - P_{el}(t) - D \cdot (\omega(t) - \omega_{ref})), \quad (14)$$

III. REAL-TIME SIGNAL MONITORING WITH LATENT TIME WINDOWS

Generative AI employs deep learning methods to automatically discover and learn patterns within input data, employing two key sub-models: the generator model, dedicated to producing new examples, and the discriminator model, focused on classifying examples as either real or fake. As depicted in Fig. 1, once a generative model is trained using data from dynamic responses of a system under various operational conditions, the pre-trained discriminator can be employed independently to validate any new signals. The model is trained to distinguish genuine system signals from manipulated data designed to deceive operators. It can be deployed in monitoring systems to detect and mitigate cyberattacks. Let $X(t)$ represent a continuous signal observed to analyze the behavior of the monitored system at time t . To effectively capture the dynamics of the monitored system, a latent time window $W(t)$ is defined as:

$$W(t) = [X(t - \tau(t)), X(t)], \quad (15)$$

The adaptive window parameter $\tau(t)$ is determined dynamically using an automated method that computes the variance of the input signal $X(t)$ over a short preliminary window. Specifically, $\tau(t) = k/\text{Var}(X(t))$, where $\text{Var}(X(t))$ is the signal's variance and k is a scaling factor calibrated from simulation testbed results. For high-variance signals, $\tau(t)$ is reduced to capture recent dynamics, while for low-variance signals, a larger $\tau(t)$ includes broader context. This prevents oversimplification from large windows, which could lead to high-risk anomaly decisions, and avoids overfitting from short windows. For new signals from different sources, $\tau(t)$ can be adjusted during training to ensure robustness across diverse signal types. The data within $W(t)$ serves as the foundation for analysis and anomaly detection, providing a real-time perspective on the system's behavior. An Anomaly Discriminator is introduced as a binary classifier to distinguish between genuine and anomalous signals. Given an input data point x within $W(t)$ at time t , the discriminator outputs a probability score

$$P_s(x) = \mathcal{D}(x, W(t), t), \quad (16)$$

where \mathcal{D} is the discriminator function. If $P_s(x)$ falls below a predefined threshold, it signals a potential anomaly or system compromise.

To enhance anomaly detection, a Generator G is employed to reconstruct input signals and assess deviations indicative of anomalies. As illustrated in the updated Fig. 2, the Generator takes as input an encoded latent representation of the input signal, produced by an encoder network E , rather than a random latent variable. Specifically, for a given time window of the input signal $W(t)$, the encoder E compresses it into a latent representation $z(t) = E(W(t))$. The Generator then reconstructs the signal from this latent input, producing an

output $\hat{W}(t)$. The reconstructed signal is given by:

$$\hat{W}(t) = G(z(t)) = G(E(W(t))). \quad (17)$$

The reconstruction error ϵ , which serves as a primary indicator for anomaly detection, is computed as the Euclidean norm of the discrepancy between the observed signal $W(t)$ and the reconstructed signal $\hat{W}(t)$:

$$\epsilon = \|W(t) - \hat{W}(t)\|_2, \quad (18)$$

where $\|\cdot\|_2$ denotes the Euclidean norm. Large values of ϵ indicate significant deviations from the learned normal data distribution, suggesting anomalous behavior. To further refine anomaly detection, the Generator's reconstruction error is combined with the Discriminator's output $D(W(t))$, which assesses the likelihood of the input signal belonging to the normal data distribution.

A. Generative Adversarial Networks (GANs) Model

The integration of next-generation clean IES faces a major challenge due to the limited availability of labeled data for anomalies and cyber threats. To address this, generative methods like GANs are utilized, offering a solution for generating realistic data in scenarios where acquiring labeled anomalous data is difficult or expensive. The GAN framework consists of two neural networks: a Generator and a Discriminator, trained in an adversarial manner. The Generator learns to create outputs that closely resemble the target data distribution. Key elements of the training process include the use of Wasserstein Distance, Lipschitz Constraint, gradient penalty, and loss functions, which are defined as follows:

1) *Wasserstein Distance*: Wasserstein distance (also known as the Earth Mover's distance) provides a continuous and gradient-based measure of the discrepancy between the generated and target distributions, instead of using a binary cross-entropy loss as in traditional GANs [22]. This allows for smoother and more robust optimization of the Generator. The formulation of the Wasserstein distance between the real data distribution P_r and the Generator's data distribution P_g in a Wasserstein GAN (WGAN) is described as follows:

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|] \quad (19)$$

where $W(P_r, P_g)$ represents the Wasserstein distance between the two distributions. \inf denotes the infimum, which is the greatest lower bound. γ is the joint distribution over x (samples from P_r) and y (samples from P_g). $\Pi(P_r, P_g)$ represents the set of joint distributions over x and y that have P_r and P_g as their marginals, respectively. $\mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|]$ denotes the expected cost of transporting mass from P_r to P_g via the coupling γ . $\|\cdot\|$ is the Euclidean norm (or L2 norm) of the vector $x - y$. In practice, this formulation can be approximated using a finite number of samples from P_r and P_g .

2) *Lipschitz Constraint*: To ensure that the Discriminator's output does not vary too wildly, thereby preventing issues such as mode collapse and vanishing gradients, we apply the Lipschitz constraint to the discriminator network. For any pair of input generated samples by generators, denoted as x_1 and

x_2 , along with their corresponding discriminator outputs $D(x_1)$ and $D(x_2)$, the Lipschitz constraint ensures that:

$$|D(x_1) - D(x_2)| \leq K \cdot \|x_1 - x_2\| \quad (20)$$

Here, K is the Lipschitz constant, which is a positive real number that restricts the rate of change of the discriminator's function, and $\|x_1 - x_2\|$ represents the distance between the input points x_1 and x_2 in the input space.

3) *Gradient Penalty*: WGAN enforces Lipschitz continuity on the discriminator by clipping its weights to a certain range $[-c, c]$. This restriction limits the Discriminator's expressiveness and ability to learn complex patterns, potentially hindering model convergence and the quality of generated samples. Finding the optimal clipping value is crucial but often challenging, as it varies depending on the dataset and model architecture. In order to enforce the Lipschitz constraint, a gradient penalty is added to the objective function [23]. This penalty encourages the gradients of the Discriminator to its input to have a norm of 1.

$$L_p = \lambda \mathbb{E}_{\hat{x} \sim P_{\hat{x}}} [(\|\nabla_{\hat{x}} D_w(\hat{x})\|_2 - 1)^2] \quad (21)$$

Here, λ is the gradient penalty coefficient controlling the strength of the penalty. Equation 21 defines the gradient penalty that enforces 1-Lipschitz continuity on the discriminator without weight clipping. The gradient penalty in WGAN-GP stabilizes training by enforcing a 1-Lipschitz continuity constraint on the Discriminator. It uses random samples $\hat{x} \sim P_{\hat{x}}$ and calculates the squared norm of the Discriminator's output gradient to \hat{x} . This approach avoids weight clipping, which can restrict model capacity, thus ensuring more stable and effective training. This is achieved by penalizing the discriminator if the gradient norm along straight lines between real and generated samples deviates from 1. To compute this penalty, points are sampled uniformly along straight lines connecting data points from the real distribution (P_r) and the Generator distribution (P_g). The ideal discriminator should maintain a gradient norm of 1 along these lines, which helps the model capture subtle patterns in both normal and anomalous data.

4) *Loss Functions*: We need to modify the loss function to be minimized by the Generator and maximized by the Discriminator by incorporating the Wasserstein distance and adding a gradient penalty to enforce a Lipschitz constraint on the discriminator. Here's the formulation for the loss function:

$$\min_{\theta} \max_w L(D_w, G_{\theta})$$

$$L(D_w, G_{\theta}) = \mathbb{E}_{x \sim P_r} [D_w(x)] - \mathbb{E}_{z \sim P_z} [D_w(G_{\theta}(z))] + L_p \quad (22)$$

where θ represents the parameters of the Generator network G , w represents the parameters of the Discriminator network D , $D_w(x)$ is the output (Wasserstein score) of the Discriminator for a real data sample x , $G_{\theta}(z)$ is the generated sample by the Generator with parameters θ , and $D_w(G_{\theta}(z))$ is the output of the Discriminator for a generated sample.

5) *The LSTM-Enhanced Generator Structure*: The Generator G_{θ} is designed as an LSTM-based network to capture temporal or sequential patterns in the data. The LSTM structure generates samples conditioned on a noise vector $z \sim P_z$,

and the output is further processed to match the dimensionality of the target distribution. The Generator is defined as:

$$G_{\theta_g}(z) = \text{Decoder}(\text{LSTM}(\text{Embedding}(z))) \quad (23)$$

The input to the Generator is an encoded noise vector $z \sim P_z$, where P_z represents a prior distribution such as Gaussian or Uniform. A fully connected embedding layer projects the noise vector z into a higher-dimensional latent space.

$$h_0 = \text{ReLU}(W_z z + b_z) \quad (24)$$

Here, W_z and b_z are learnable parameters, and h_0 serves as the initial hidden state for the LSTM. To incorporate autocorrelation structure, lagged hidden states are explicitly included in the LSTM's input. At each timestep t , the LSTM receives a concatenation of the previous hidden state h_{t-1} and a context vector ρ_t capturing autocorrelated features from prior lags $\ell \in \mathcal{L}$:

$$\rho_t = \sum_{\ell \in \mathcal{L}} \alpha_{\ell} h_{t-\ell} \quad (25)$$

Here, α_{ℓ} are learnable coefficients (or fixed based on empirical autocorrelation values), and \mathcal{L} denotes the set of significant lags based on the Autocorrelation Function. The LSTM then updates its hidden and cell states as follows:

$$h_t, c_t = \text{LSTM}(\text{concat}(h_{t-1}, \rho_t), c_{t-1}; \theta_{\text{LSTM}}) \quad (26)$$

This structure allows the network to explicitly consider dependencies on multiple past time steps, as suggested by autocorrelation patterns. The output of the LSTM is passed through a decoder that maps it back into the data space:

$$\hat{x} = \text{Tanh}(W_h h_T + b_h) \quad (27)$$

Here, W_h and b_h are learnable parameters, h_T is the final hidden state of the LSTM, and \hat{x} is the Generated data sample. The parameters θ of the Generator are optimized to minimize the loss function in conjunction with the Discriminator. The overall objective encourages G_{θ} to generate samples indistinguishable from real data under the Discriminator's scoring mechanism.

B. Generative Model in Cybersecurity

We chose the LSTM-GAN framework because it effectively models non-stationary time series with long-term dependencies, leveraging LSTM's ability to capture sequential patterns that Transformer-based models lack. Additionally, the GAN discriminator naturally provides a probabilistic measure of how likely a sequence is real or fake, which is essential for our anomaly detection task. Compared to Transformer models, LSTM-GANs require less data and offer more stable training, making them well-suited for generating realistic sequences and supporting reliable probabilistic anomaly detection in our setting. In Algorithm 1, the process is divided into two phases: training and testing. First, real data $\{x_t\}$ is sampled from the true distribution $\mathcal{X}_{\text{real}}$, and noise vectors $\{z_t\}$ are sampled from a prior distribution. The Discriminator D_w is trained to minimize the Wasserstein distance between the real and fake data $D_w(x_t)$ and $D_w(\hat{x}_t)$, with a gradient penalty λ to stabilize training. The Discriminator's parameters w are updated to minimize

Algorithm 1 Real-Time Anomaly Detection Algorithm

Require: • Training data $\mathcal{X}_{\text{train}} = \{x_t\}$

- Hyperparameters: Learning rate η , batch size B , training epochs E
- Model architecture: LSTM-Generator $G_{\theta}(z)$ and GP-Discriminator $D_w(x_t)$

- 1: **Initialize:** Define model architectures for G_{θ} and D_w .
- 2: Randomly initialize weights θ for G_{θ} and w for D_w .

3: **Training Phase:**

- 4: **for** $e = 1$ to E (Training Epochs) **do**

5: **for** each batch **do**

- 6: Sample real data $\{x_t\}_{t=1}^B$ from $\mathcal{X}_{\text{train}}$.

- 7: Sample noise vectors $\{z_t\}_{t=1}^B$ from prior distribution $p(z)$.

- 8: Generate fake samples: $\hat{x}_t = G_{\theta}(z_t)$.

- 9: Compute discriminator loss:

$$\mathcal{L}_D = \mathbb{E}[D_w(\hat{x}_t)] - \mathbb{E}[D_w(x_t)] + \lambda \cdot \text{GP}$$

- 10: Update discriminator weights: $w \leftarrow w - \eta \nabla_w \mathcal{L}_D$.

- 11: Compute generator loss:

$$\mathcal{L}_G = \lambda_1 \|x_t - \hat{x}_t\|_2 + \lambda_2 \cdot (-\mathbb{E}[D_w(\hat{x}_t)])$$

- 12: Update generator weights: $\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}_G$.

13: **end for**

14: **end for**

15: **Real-Time Detection Phase:**

- 16: **for** each incoming signal x_t **do**

- 17: Compute discriminator score: $P_t = D_w(x_t)$.

- 18: Compute predictive error: e_t .

- 19: If P_t or e_t exceeds threshold, flag anomaly.

20: **end for**

the total loss, and the Generator's parameters θ are updated to minimize the negative Wasserstein distance $-D_w(G_{\theta}(z))$ and the prediction loss. After the training phase, where both the Generator and Discriminator are optimized, the pre-trained model is then used in the testing phase.

In the real-time detection phase, each incoming signal x_t is processed by the Discriminator to obtain a probability score $P_t = D_w(x_t)$, while the Generator predicts and calculates the error between expected and observed signals in the next latent windows e_t . To generate a final probability score, the model compares the prediction error e_t with the discriminator score P_t . If the score exceeds a threshold, x_t is flagged as anomalous; otherwise, it is marked as normal. The thresholds for anomaly detection are chosen based on the required sensitivity, where lower thresholds increase sensitivity and false positives, while higher thresholds reduce sensitivity and false positives. The scenario illustrated in Fig. 2 depicts a cyberattack known as "Data Manipulation" or "Data Tampering," where real-time monitoring data is replaced or injected with false information to mislead or disrupt decision-making processes.

The proposed anomaly detection model addresses this threat by leveraging a Generator and a Discriminator trained on historical system behavior to identify deviations from normal operation. After training, a combination of real and generated signals is used to evaluate the Discriminator and Generator.

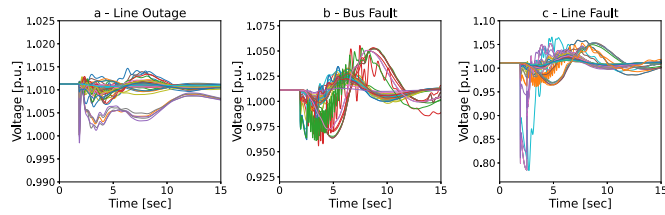


Fig. 3. Dynamic voltage responses in various fault scenarios represented in the real dataset.

Incoming signals are divided into windows, and each is analyzed using a moving window approach. Anomaly likelihood is assessed using the discriminator's score and the predictive error. A threshold is established to classify signals, and scores across windows are aggregated for a final decision.

IV. NUMERICAL RESULTS

A. Dataset and Training Process

A database for model training is developed using simulations in Power System Simulation for Engineers (PSSE). A detailed dynamic model of the IES and its connection to the main grid is created, incorporating diverse operational scenarios to reflect realistic grid conditions. The IES includes components such as a battery energy storage system, wind, solar, and thermal generators. Two IES models are connected to the IEEE 118-bus system at bus-111 and bus-54. Simulations encompass various faults and outages, including generator trips, line and transformer faults, equipment failures, extreme weather impacts, and cyberattacks, characterized by fault location, magnitude, and duration. Time-domain simulations generate data on parameters like voltage, power generation, frequency deviations, grid conditions, and control actions. This data is organized into a database to facilitate model training. The primary concern revolves around distinguishing signals received at the interconnection point (bus-111) as either resulting from the system's normal response or if monitoring signals have been replaced by fraudulent signals, to mislead the operator into making erroneous decisions when there is no real issue. Random events like line outages, faults, and bus trips are simulated in the interconnected grid to create a dataset capturing the system's response, assuming a constant thermal load.

The results of voltage monitoring under various disruption scenarios are illustrated in Fig. 3. The random outage events are generated using a Monte Carlo simulation, where lines and buses in the main grid are randomly selected to experience faults. Each scenario is simulated over a 15-second dynamic time window, with the random outage occurring at $t=2$ seconds. This approach ensures a statistically broad range of possible failure events that reflect potential real-world disturbances. (a) Voltage responses at the IES connection point to the grid during 40 random line outage events. (b) Voltage profiles at the connection point during 40 random bus faults in the main grid. (c) Voltage responses during 40 random three-phase-to-ground line faults, with all faults cleared after

TABLE I
TRAINING DETAILS

Parameter	Value
Optimizer	RMSprop
Total steps (N_{total})	3500
Rollout buffer size (N_{steps})	5e4
Batch size (N_{batch})	40
Learning rate	1e-4
Length of noise	100
Epochs	35

0.2 second. The dynamic model of the IES incorporates different modules, including the Generator/Converter module (REGCA), Plant Controller module (REPC), and Electrical Control modules (REECC for battery, REECA for wind, and REECB for solar). While the dataset used in this work is generated via high-fidelity nonlinear dynamic simulations in PSS® E, future extensions may incorporate field data to further validate and enhance the AI model's robustness under real-world conditions.

B. Implementation and Detection

In the WGAN-GP algorithm, the gradient penalty is introduced to enforce a Lipschitz constraint on the Discriminator, preventing it from undergoing overly steep changes. This is achieved by interpolating randomly sampled values ($\varepsilon \sim \text{Uniform}(0, 1)$) between real and fake data points.

$$x_p = \varepsilon \cdot \text{real_data} + (1 - \varepsilon) \cdot \text{fake_data} \quad (28)$$

The gradient of the Discriminator's output to the interpolated data is used as a penalty term, encouraging gradient norms near 1 for stable and meaningful learning. This enhancement contributes to the effectiveness of the WGAN-GP training process in generating high-quality samples. The different parameters set after an empirical tuning process are presented in Table I. In this work, the Root Mean Square Propagation (RMSprop) optimizer is selected due to its effectiveness in handling non-stationary objectives and sparse gradients, which are common in the training of neural networks modeling complex and coupled systems like the Integrated Energy System (IES). RMSprop adapts the learning rate individually for each parameter based on a moving average of recent gradient magnitudes. This is particularly useful for IES applications, where the dynamics of multiple energy carriers (electricity, thermal, etc.) can vary significantly over time and across subsystems. By normalizing the updates using the decayed average of past squared gradients, RMSprop helps stabilize training and mitigates vanishing/exploding gradient issues in deeper architectures. Additionally, to reflect the multi-scale and interactive features of IES, we adjusted the learning rate and decay parameters in RMSprop to better capture the slower thermal responses and faster electrical dynamics during training.

The Discriminator loss measures its ability to differentiate real from generated samples. As shown in Fig. 4, the loss initially decreases from 0 to -0.2 in the first 500 epochs, indicating the Generator's improved sample quality. By epoch

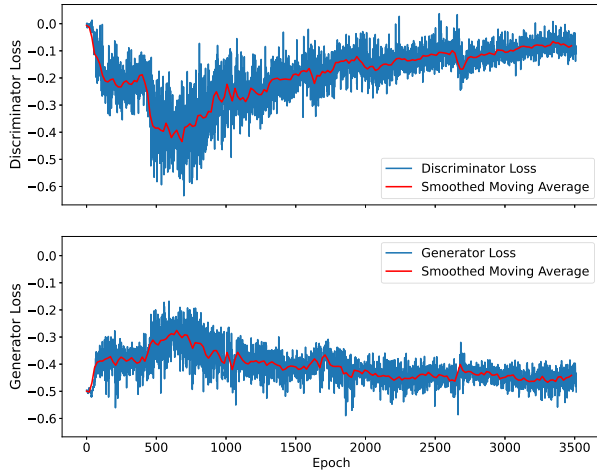


Fig. 4. Discriminator and generator losses during training over epochs.

1000, the loss drops to -0.6 , reflecting the Generator's success in producing more challenging samples. The Generator loss increases over time, signifying its growing difficulty in fooling the Discriminator, eventually stabilizing around -0.4 . This balance marks an equilibrium where the Generator's samples become nearly indistinguishable from real data. The negative loss values result from the Wasserstein loss, which measures the difference between real and generated data distributions. The model runs efficiently on standard CPU hardware and is compatible with GPUs if available, without requiring infrastructure upgrades. In this study, all experiments were performed on CPU hardware. All model training and computational experiments were conducted on a workstation equipped with an Intel® Xeon® E5-2603 v3 CPU with 12 cores.

Post-training the GAN model, the pre-trained discriminator is tested with a new set of fake signals designed to emulate cyberattack scenarios that induce system instability. As illustrated in Fig. 5, the discriminator's probability score for cyberattack signals consistently drops within a specific window, indicating the presence of anomalies.

$$D_w(x, \epsilon) = P_r(\text{real}|x) \cdot \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)} \odot (1 - \epsilon) \quad (29)$$

The equation for $D_w(x, \epsilon)$ integrates the Discriminator function $D_w(x)$ with the predictive error of the generator ϵ to enhance the identification of anomalies. The term $P_r(\text{real}|x)$ represents the Discriminator's probability that a given signal x is real, while the fraction $\frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$ balances the real data distribution with the generated data distribution. The addition of $1 - \epsilon$ is the predictive error, as a penalty for high discrepancies between predicted and observed signals. Larger values of ϵ (indicating greater deviation or anomaly) decrease the probability score, signaling that the input x is more likely to be anomalous. An anomaly is flagged when this combined probability score $D_w(x, \epsilon)$ is close to zero, indicating both low confidence that the signal is real and a high reconstruction error. The modification enables the Discriminator to detect

anomalies by identifying data that significantly deviates from expectations, in addition to distinguishing real from generated data.

As depicted in Fig. 5, three representative attack scenarios were constructed using techniques such as signal shifting and blending, variance manipulation, and oscillation enhancement. For each incoming signal in the monitoring system, the proposed model applies a sliding window to compute both the Discriminator's probability score and the Generator's predictive error. The final anomaly score is derived by combining these two metrics. The green-shaded areas represent the Discriminator's probability scores. When the Discriminator's output for a given window falls below 0.5, it suggests that the observed signal deviates from patterns the model has learned during training. The blue curve overlaid on the figure illustrates the Generator's predictive error, representing the difference between the actual and predicted signals. A low Discriminator score, combined with a high Generator error, signals a higher likelihood of an anomaly within the evaluated signal window.

To assess the effect of the detection threshold, we conducted a sensitivity analysis using 50 fake samples. Lowering the detection threshold increased fake sample detection from 12% (threshold 0.7) to 46% (0.4) and 93% (0.1), illustrating the trade-off between sensitivity and false alarms.

C. Case 1: Model Testing With a Constant Thermal Load

Before testing, the training was evaluated to assess how well the model had been trained. To this aim, the probability distributions of the generated and real datasets have been compared. This is often represented by a probability density function (PDF), which indicates the likelihood of various outcomes for a random variable. In Fig. 6, kernel density plots are used to contrast the generated signals with the real dataset. The significant overlap between the blue curve (generated signals) and the orange curve (real dataset) indicates a high similarity in the distribution of values. The consistent spread of the curves further suggests that the variability in the generated signals aligns well with that of the real dataset. The dashed-line area represents the training of the model with a Deep Convolutional (DC)-GAN model. Simpson's rule calculates the intersection area between the models. The intersection of the trained model with GAN with the real data set is 71.16%, while the intersection of WGAN-GP is 89.87% and LSTM-GAN is 93.28%. The similarity in the shapes of the curves underscores how closely the generated signals follow the underlying patterns in the actual data. These observations suggest that the generated signals effectively mirror the distributional characteristics of the real dataset in trained networks.

We designed our approach to extract generalizable features of system dynamics, such as response shape, damping behavior, temporal consistency, and spatial correlations, instead of learning attack-specific patterns. This design enables the model to maintain strong detection performance even when confronted with previously unseen attack types, even without explicit training on them.

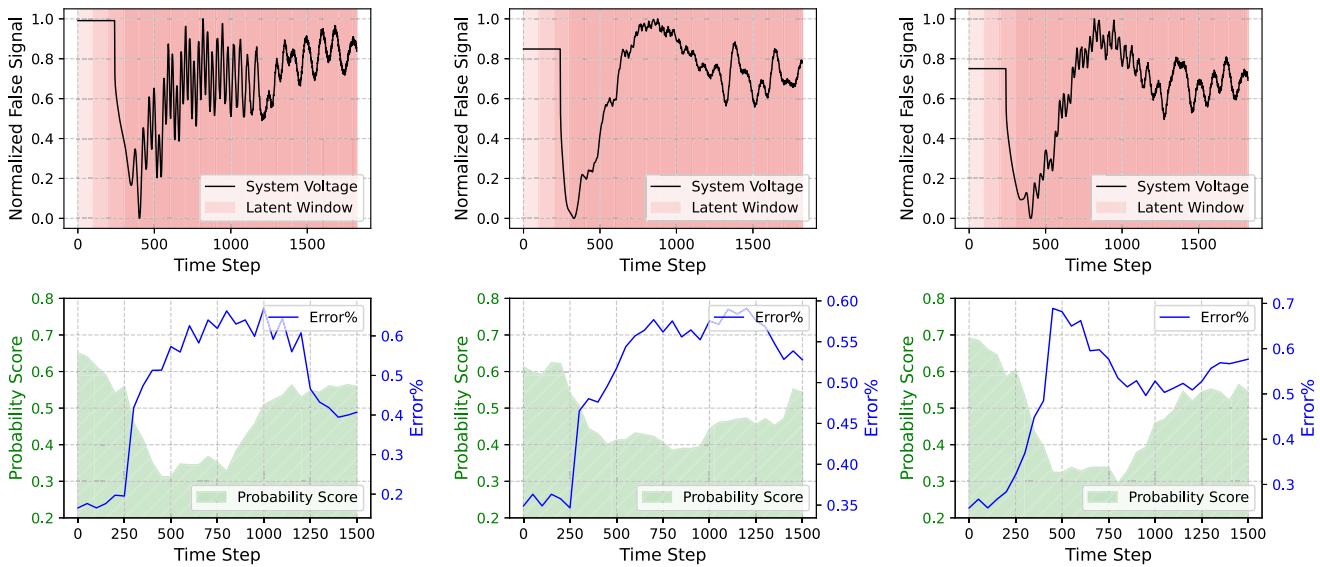


Fig. 5. Anomaly detection by utilizing a probability score calculated from the output of the pre-trained discriminator.

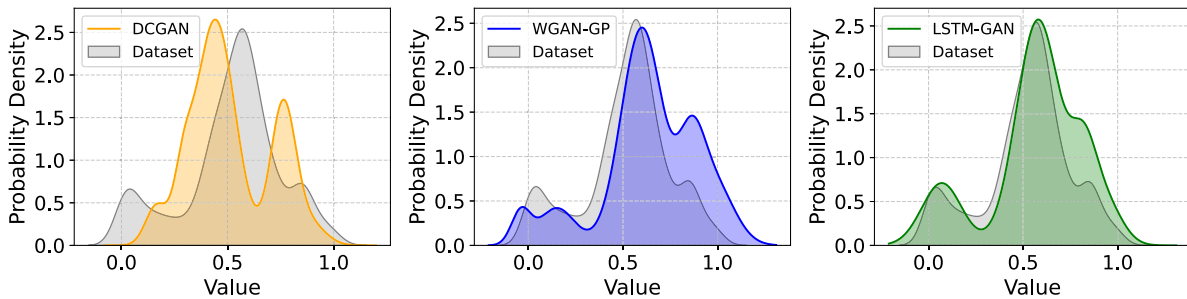


Fig. 6. Comparing real and synthetic data distributions: LSTM-GAN (green) and WGAN-GP (blue) and DCGAN (yellow), with real data in gray.

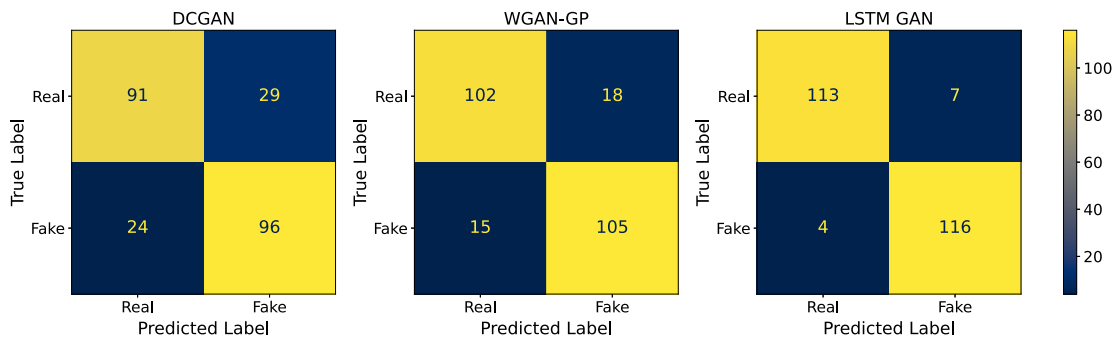


Fig. 7. Comparative confusion matrices for analyzing the accuracy, precision, and recall of detection methods. (total samples = 240; 120 samples per class).

To achieve this, the training phase relies exclusively on simulation-based data, ensuring the model learns clean, unbiased representations of physical fault responses under controlled and accurately labeled conditions. This approach facilitates generalization from fundamental system dynamics rather than overfitting to particular scenarios. For evaluation, we use a separate test set containing both real and adversarially manipulated data. The attack scenarios included in testing span a range of cyber-attack strategies, such as time-shifted measurements (shift attacks), artificially altered measurement

volatility (variance manipulation), amplified dynamic behavior (oscillation enhancement), injected signals from unrelated systems (cross-system data injection), mixtures of real and synthetic data (signal blending), structured and unstructured noise injection, and targeted manipulation of subsets of sensor measurements (partial spoofing).

The evaluation dataset consists of 240 cases (120 Real, 120 Fake), deliberately constructed to span a diverse set of representative cyber-attack strategies. The balanced design ensures reliable estimation of classification performance and

TABLE II
PERFORMANCE COMPARISON OF METHODS

Metric	DCGAN	WGAN-GP	LSTM GAN
Accuracy	0.779	0.862	0.954
Precision	0.768	0.853	0.943
Recall	0.800	0.875	0.966
F1 Score	0.783	0.863	0.955
Type 1 Error	0.241	0.150	0.058
Type 2 Error	0.200	0.125	0.033

avoids bias from class imbalance, while the diversity of cases makes the dataset sufficiently comprehensive to demonstrate the model's generalization capability within the scope of this study. As shown in Fig. 7, anomaly detection using a pre-trained DCGAN model correctly identified 96 fake cases (True Positives, TP1) but misclassified 24 fake cases as real (False Negatives, FN1). Additionally, the DCGAN model accurately identified 91 real cases (True Negatives, TN1) but wrongly classified 29 real cases as fake (False Positives, FP1). This yielded an accuracy of $(TP1 + TN1)/(TP1 + FN1 + FP1 + TN1) = 0.779$, a precision of $TP1/(TP1 + FN1) = 0.768$, and a recall of $TP1/(TP1 + FP1) = 0.80$. Evaluating Type I and Type II errors offers deeper insight into the model's performance beyond basic accuracy. Type I error (False Positive Rate) measures the proportion of fake cases incorrectly labeled as real, which is important in scenarios where false alarms have costly or disruptive consequences. Type II error (False Negative Rate) reflects the proportion of real cases misclassified as fake, which is significant when missing real cases could lead to serious consequences. Method 1's Type I error rate is $FP1/(FP1 + TN1) = 0.241$, while its Type II error rate is $FN1/(FN1 + TP1) = 0.20$.

In the same way, as shown in Table II, using WGAN-GP models, correctly classified 105 fake cases (TP2), with only 15 misclassified as real (FN2). It also accurately identified 102 real cases (TN2), with just 18 incorrectly labeled as fake (FP2). This resulted in a higher accuracy of 0.862, a precision of 0.853, and a recall of 0.875.

Finally, the LSTM-GAN enhanced models correctly classified 116 fake cases (TP3), with only 4 misclassified as real (FN3). It also accurately identified 113 real cases (TN3), with just 7 incorrectly labeled as fake (FP3). This resulted in a significantly higher accuracy of 0.954, a precision of 0.943, and a recall of 0.966.

The proposed LSTM-GAN enhanced model, which incorporates the predictive error of the generator and the discriminator probability scores from WGAN-GP, consistently outperforms the initial DCGAN model across all key metrics. It achieves a significantly higher accuracy of 95%, compared to 77% for the DCGAN model, in correctly classifying cases. The proposed LSTM-GAN enhanced model's precision of 94% demonstrates its ability to identify real cases more accurately while reducing false positives. Additionally, its recall of 96% shows a greater ability to capture real cases, minimizing false negatives. The large reduction in Type I and Type II error rates further highlights the method's effectiveness.

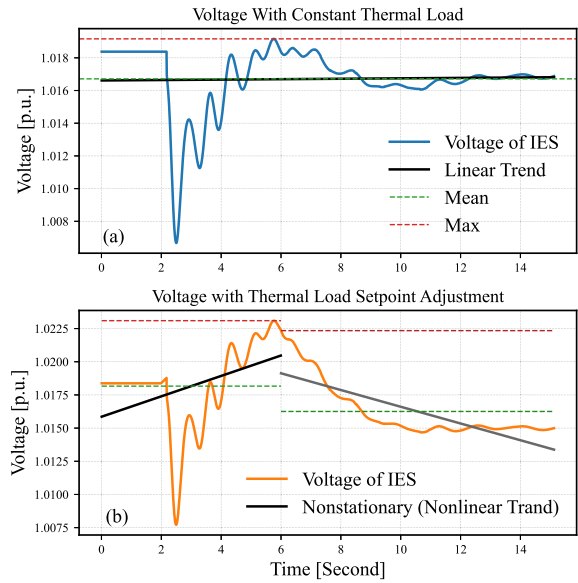


Fig. 8. Voltage at the IES-grid connection point during a grid event: (a) with constant thermal load (stationary behavior), and (b) with changing thermal load setpoint (non-stationary response).

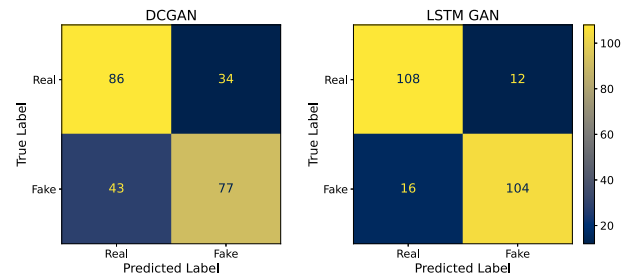


Fig. 9. Comparison of confusion matrices for anomaly detection in IES with variable thermal loads.(total samples = 240; 120 samples per class).

TABLE III
MODEL PARAMETERS FOR SYSTEM DYNAMICS

Parameter	Value	Unit	Explanation
τ_{set}	5.0	s	Thermal setpoint time constant
$P_{th,max}$	45.0	MW_{th}	Max thermal power
$P_{el,max}$	30.0	MW_e	Max electrical power
k	1.5	–	Thermal-electrical scaling
τ	10.0	s	Thermal power time constant
τ_g	0.4	s	Governor time constant
K_d	10.0	–	Electrical damping gain
ω_{ref}	1.0	p.u.	Rotor speed reference
H	6.5	s	Inertia constant
D	0.5	–	Rotor damping coefficient

D. Case 2: IES With Variable Thermal Load

In this case, we simulate the dynamic behavior of thermal loads that interact with the electrical system through combined heat and power generators. The thermal load's dynamic response is modeled with a user-defined PSS® E model, enabling coupling with the electrical system. The IES and grid configuration remain the same; however, instead of a constant thermal load, we vary the thermal load setpoints

to assess their dynamic impact on electrical parameters. As shown in Fig. 8(a), when the thermal load remains constant over time, the voltage at the connection point between the IES and the main grid, during a random event in the main grid, exhibits stationary behavior. This means that the statistical properties of the voltage signal, such as its mean and variance, do not change over time, allowing conventional analysis and detection methods to perform reliably. In contrast, as shown in Fig. 8(b), when the thermal load setpoint changes (specifically at time 6, from 42 MW_{th} to 25 MW_{th}), the inherently slow and time-varying nature of thermal demand introduces a distinctive non-stationary pattern in the voltage signal. This behavior is unique to IES systems due to their coupling between thermal and electrical dynamics. The change in the thermal load alters the statistical characteristics of the electrical signals, transitioning them from stationary to non-stationary. That is, the signal's mean, variance, and autocorrelation evolve, reflecting the long-memory and low-frequency dynamics introduced by the thermal system. Unlike typical electrical loads that cause rapid transients without significantly altering the underlying statistical properties of the signal, thermal loads introduce slow, persistent changes that gradually shift the system's behavior. These changes affect key statistical features such as the mean, variance, and autocorrelation structure of electrical signals, leading to a transition from stationary to non-stationary behavior. This non-stationarity poses a significant challenge to conventional monitoring and detection techniques, which often rely on the assumption that system signals remain statistically stable over time. Therefore, to accurately capture and respond to these evolving dynamics, there is a need for advanced generative AI models for complex, time-dependent patterns and an effective model for non-stationary behavior. The parameters for the variable thermal load integrated system are presented in Table III. By introducing this new setting where the thermal load setpoint changes at time 6, we generated a novel dataset consisting of voltage measurements at the connection point, recorded during various random events in the main grid while the thermal setpoint was dynamically varying. This dataset captures the complex interactions and non-stationary behaviors induced by the changing thermal load. After training, we constructed a new dataset consisting of 120 real samples obtained from simulations, similar to the first case study, and 120 synthetic attack samples generated based on a diverse set of cyber-attack strategies. The detection results are summarized in the confusion matrix shown in Fig. 9, where the performance of our proposed model is compared against the benchmark DCGAN model. The model achieves 88.33% accuracy and 89.65% precision, outperforming the baseline by effectively capturing temporal dependencies and non-stationary behavior, demonstrating the advantage of temporal memory for anomaly detection in coupled thermal–electrical systems.

V. CONCLUSION

The integration of IES with power grids is vital for enhancing the efficiency, security, and resiliency of modern energy management systems. However, the growing interconnectivity of energy sources, smart grids, and digital control systems

exposes IES to significant cyber threats. To address this, we developed a real-time AI-assisted monitoring framework for anomaly detection, leveraging generative AI models to monitor grid-connected IES. We propose a combination of a WGAN-GP and LSTM model, which excels in feature extraction from high-similarity datasets. This approach allows the system to adapt to evolving patterns and significantly enhances anomaly detection performance. Our methodology uses predictive modeling with an enhanced discriminator probability score from a pre-trained Generator and Discriminator to identify anomalies in real time. The method demonstrates improved performance, achieving 95% overall accuracy and 94% precision, meaning that 94% of detected anomalies are correctly identified. Furthermore, it maintains a 96% recall rate, successfully identifying 96% of all actual attack signals.

Future work could integrate the model with a digital twin or intelligent agent for rapid anomaly detection and adaptive responses, creating a self-learning, resilient system with continuous situational awareness. Additionally, incorporating Graph Convolutional Networks (GCNs) to represent the system's topology enhances the model's ability to identify system vulnerabilities, support edge conditions, and address system-level stress and low-intensity cyberattacks through more effective cyber-defense strategies. When an anomaly is detected, the system can also prompt Generative Pre-trained Transformer-based agents to generate a human-readable root cause analysis with enhanced interpretability, identify probable causes, and suggest corrective actions for the grid operators.

DATA AVAILABILITY

The code and data that support the findings of this study are available in the GitHub repository at github.com/UTDDOES/GAN-model-for-Anomaly-Detection.

REFERENCES

- [1] J. Jia, H. Li, D. Wu, J. Guo, L. Jiang, and Z. Fan, "Multi-objective optimization study of regional integrated energy systems coupled with renewable energy, energy storage, and inter-station energy sharing," *Renew. Energy*, vol. 225, May 2024, Art. no. 120328.
- [2] D. J. Arent et al., "Multi-input, multi-output hybrid energy systems," *Joule*, vol. 5, no. 1, pp. 47–58, 2021.
- [3] Y. Wang, J. Hu, and N. Liu, "Energy management in integrated energy system using energy-carbon integrated pricing method," *IEEE Trans. Sustain. Energy*, vol. 14, no. 4, pp. 1992–2005, Oct. 2023.
- [4] M. Zhu et al., "A comprehensive methodology for optimal planning of remote integrated energy systems," *Energy*, vol. 285, Dec. 2023, Art. no. 129443.
- [5] M. Chaudry, L. Jayasuriya, J. W. Hall, N. Jenkins, N. Eyre, and S. Eggimann, "Simulating flexibility, variability and decentralisation with an integrated energy system model for great Britain," *Sci. Rep.*, vol. 13, no. 1, p. 4772, Mar. 2023.
- [6] C. Li et al., "Optimal planning of islanded integrated energy system with solar-biogas energy supply," *IEEE Trans. Sustain. Energy*, vol. 11, no. 4, pp. 2437–2448, Oct. 2020.
- [7] T. Jiang, T. Sun, G. Liu, X. Li, R. Zhang, and F. Li, "Resilience evaluation and enhancement for island city integrated energy systems," *IEEE Trans. Smart Grid*, vol. 13, no. 4, pp. 2744–2760, Jul. 2022.
- [8] J. Rahman, R. A. Jacob, and J. Zhang, "Multi-timescale power system operations for electrolytic hydrogen generation in integrated nuclear-renewable energy systems," *Appl. Energy*, vol. 377, Jan. 2025, Art. no. 124346.
- [9] M. Yan, Y. He, M. Shahidepour, X. Ai, Z. Li, and J. Wen, "Coordinated regional-district operation of integrated energy systems for resilience enhancement in natural disasters," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 4881–4892, Sep. 2019.

- [10] S. Ding, W. Gu, S. Lu, R. Yu, and L. Sheng, "Cyber-attack against heating system in integrated energy systems: Model and propagation mechanism," *Appl. Energy*, vol. 311, Apr. 2022, Art. no. 118650.
- [11] W. Guo, S. Sun, C. Tang, G. Li, X. Bai, and Z. Zhao, "Classification of anomaly patterns in integrated energy systems based on conditional variational autoencoder and attention mechanism," *Energies*, vol. 16, no. 11, p. 4367, May 2023.
- [12] L. Zhang, H. Su, E. Zio, L. Jiang, L. Fan, and J. Zhang, "A graph structure feature-based framework for the pattern recognition of the operational states of integrated energy systems," *Expert Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 119039.
- [13] J. Chen, S. Zhou, Y. Qiu, and B. Xu, "An anomaly detection method of time series data for cyber-physical integrated energy system based on time-frequency feature prediction," *Energies*, vol. 15, no. 15, p. 5565, Jul. 2022.
- [14] L. Cai, N. F. Thornhill, S. Kuenzel, and B. C. Pal, "Wide-area monitoring of power systems using principal component analysis and k -nearest neighbor analysis," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 4913–4923, Sep. 2018.
- [15] M. Ganjkhani, S. N. Fallah, S. Badakhshan, S. Shamshirband, and K.-W. Chau, "A novel detection algorithm to identify false data injection attacks on power system state estimation," *Energies*, vol. 12, no. 11, p. 2209, Jun. 2019.
- [16] J. Wu, X. Chen, S. Badakhshan, J. Zhang, and P. Wang, "Spectral graph clustering for intentional islanding operations in resilient hybrid energy systems," *IEEE Trans. Ind. Informat.*, vol. 19, no. 4, pp. 5956–5964, Apr. 2023.
- [17] M. J. Pappaterra and F. Flammini, *Bayesian Networks for Online Cybersecurity Threat Detection*. Cham, Switzerland: Springer, 2021, pp. 129–159.
- [18] G. Li and J. J. Jung, "Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges," *Inf. Fusion*, vol. 91, pp. 93–102, Mar. 2023.
- [19] K. Ren, Y. Zeng, Z. Cao, and Y. Zhang, "ID-RDRL: A deep reinforcement learning-based feature selection intrusion detection model," *Sci. Rep.*, vol. 12, no. 1, p. 15370, Sep. 2022.
- [20] J. Zhao, F. Li, H. Sun, Q. Zhang, and H. Shuai, "Self-attention generative adversarial network enhanced learning method for resilient defense of networked microgrids against sequential events," *IEEE Trans. Power Syst.*, vol. 38, no. 5, pp. 4369–4380, Sep. 2023.
- [21] C. Ren and Y. Xu, "A fully data-driven method based on generative adversarial networks for power system dynamic security assessment with missing data," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 5044–5052, Nov. 2019.
- [22] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 70, D. Precup and Y. W. Teh, Eds., PMLR, Aug. 2017, pp. 214–223. [Online]. Available: <https://proceedings.mlr.press/v70/arjovsky17a.html>
- [23] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," 2017, *arxiv:1704.00028*.