

Article

Time-Series-Based Anomaly Detection in Industrial Control Systems Using Generative Adversarial Networks

Chungku Han ¹ and Gwangyong Gim ^{2,*} 

¹ Department of IT Policy and Management, Soongsil University, Seoul 06978, Republic of Korea; dooli29@gmail.com

² Department of Business Administration, Soongsil University, Seoul 06978, Republic of Korea

* Correspondence: gygim@ssu.ac.kr

Abstract

Recent advances in time-series anomaly detection have leveraged artificial intelligence (AI) to improve detection performance. In industrial control systems (ICSs), however, acquiring training data is challenging due to operational constraints and the difficulty of system shutdowns. To address this, many countries are developing ICS simulators and testbeds to generate training data. This study uses a publicly available ICS testbed dataset as a benchmark for the discriminator in a Semi-Supervised Generative Adversarial Network (SGAN). The goal is to generate large volumes of synthetic time-series data through adversarial training between generator and discriminator networks, thereby mitigating data scarcity in ICS anomaly detection. Comparative experiments were conducted using this synthetic data to evaluate its impact on existing detection models. Using the HAI 22.04 dataset from the National Security Research Institute, this study performed feature engineering and preprocessing to identify correlations and remove irregularities. Various models, including One-Class SVM, VAE, CNN-GRU-Autoencoder, and CNN-LSTM-Autoencoder, were trained and tested on the dataset. A synthetic dataset was then generated via SGAN and validated using PCA and t-SNE. The results show that applying SGAN-generated data to time-series anomaly detection yielded significant performance improvements in F1 score. Additional validation using the SWaT dataset from the National University of Singapore confirmed similar gains. These findings indicate that synthetic data generated by SGANs can effectively enhance semi-supervised learning for anomaly detection, classification, and prediction in data-constrained environments such as medical, industrial, transportation, and environmental systems.

Keywords: machine learning; deep learning; time-series analysis; convolutional neural network; CNN; LSTM; CNN-LSTM; SGAN



Academic Editor: Changhee Kim

Received: 1 July 2025

Revised: 4 September 2025

Accepted: 8 September 2025

Published: 9 September 2025

Citation: Han, C.; Gim, G.

Time-Series-Based Anomaly Detection in Industrial Control Systems Using Generative Adversarial Networks. *Processes* **2025**, *13*, 2885. <https://doi.org/10.3390/pr13092885>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

While the scope of applying IT technologies in industrial control systems (ICSs) is rapidly expanding, the incidence of cyberattacks is also rising—such as the hacking of major national ICS facilities in the United States (2015), the Kyiv blackout in Ukraine (2016), and the hacking of a chemical plant in Saudi Arabia (2018). Given the critical role of ICS, establishing a secure operational environment is essential. Accordingly, a growing body of research focuses on detecting abnormal behaviors in ICS [1].

The industrial control system (ICS) is an essential component in the management and control of industrial processes where security and reliability are of utmost importance.

Anomaly detection is an essential mechanism for detecting deviations from normal system behavior and identifying potential attacks or system failures. However, generating labeled datasets for supervised machine learning models is a challenging task in ICSs because it requires significant effort and expertise [2].

Recent studies on anomaly detection in time-series have also utilized publicly available industrial datasets. For example, Hoh et al. (2022) proposed a generative model that classifies normal and abnormal signals using raw audio data from the MIMII dataset [3].

In particular, research on identifying abnormal conditions in ICS operational signals using artificial intelligence (AI) is actively progressing, with recent efforts centered on deep learning-based anomaly detection models. For AI to effectively detect anomalies in ICSs, securing appropriate datasets for model training is vital. To address this challenge, countries around the world are employing simulation platforms to generate ICS-related AI training datasets.

Manipulating live ICS environments to collect anomaly detection data is virtually impossible. Therefore, many institutions build isolated control environments that mimic real ICS operations to generate the required datasets through controlled experiments. For example, the iTrust laboratory at the Singapore University of Technology and Design (SUTD) has developed testbeds for water treatment, drainage systems, and power control systems to provide such datasets. Among the most well-known is the Secure Water Treatment (SWaT) dataset.

As more datasets suitable for AI model training become available in the ICS domain, AI-based anomaly detection research has accelerated. Since access to high-quality training data remains a central challenge in AI model development, many prior studies have addressed data scarcity. Similarly, in the medical domain, researchers face limitations in data availability due to legal and ethical constraints, such as privacy protection laws. In response, recent research has increasingly focused on using generative AI to synthesize artificial medical data for use in prediction and classification models [4].

This study proposes a method to mitigate data scarcity in ICS anomaly detection by constructing a Semi-Supervised Generative Adversarial Network (SGAN), an enhanced form of the widely used Generative Adversarial Network (GAN) for synthetic data generation. The SGAN framework aims to augment ICS training datasets with synthetic data and evaluates its effectiveness by applying it to time-series-based deep learning models. This study compares models trained solely on real data with those trained on a combination of real and SGAN-generated data to assess improvements in anomaly detection performance.

This paper presents a method for generating time-series ICS data using SGAN, a semi-supervised learning framework, to compensate for limited training data. The quality of the synthetic data is validated through two techniques: (1) Principal Component Analysis (PCA) and (2) distance-based similarity assessments. Based on these results, the validated synthetic dataset is applied to hybrid deep learning models to evaluate performance improvements.

Widely used ICS simulation datasets are employed in this study. These include the HAI dataset released by South Korea's National Security Research Institute and the SWaT dataset developed by SUTD's iTrust research center. Feature engineering is conducted through data collection and preprocessing, and key features influencing anomaly detection are identified using PCA. Dimensionality reduction is also applied to observe its effect on model performance. Statistical and machine learning methods for ICS anomaly detection are compared with deep learning models trained on GAN-augmented datasets to assess the significance of performance enhancements.

To compare anomaly detection performance, the study applies deep learning-based models to SGAN-generated synthetic time-series datasets. The evaluated models include

the One-Class Support Vector Machine (OCSVM), Long Short-Term Memory Variational Autoencoder (LSTM-VAE), CNN-GRU-Autoencoder, and CNN-LSTM-Autoencoder. The quality of SGAN-generated synthetic datasets is assessed against real datasets using PCA and t-distributed Stochastic Neighbor Embedding (t-SNE). Ultimately, hybrid deep learning models trained on SGAN-generated data are compared with those trained solely on limited real data from physical testbeds (HAI and SWaT). This comparison verifies the contribution of SGAN-generated synthetic ICS time-series data to model performance, relative to existing deep learning, statistical, and machine learning methods [5–12].

This study contributes by leveraging SGAN-generated synthetic time-series data to enhance the performance of multiple anomaly detection models in ICSs, demonstrating improvements across diverse architectures. Unlike prior studies that relied solely on limited real-world ICS datasets, this study introduces a Semi-Supervised GAN (SGAN) framework to generate synthetic time-series data, thereby directly addressing the persistent issue of data scarcity in ICS anomaly detection. The proposed method is novel in three aspects: (1) it applies a semi-supervised learning paradigm to ICS data synthesis, an approach rarely explored in this domain; (2) it validates the quality of synthetic data using multiple statistical and visualization-based techniques, including PCA, t-SNE, and distance-based similarity assessments; and (3) it empirically demonstrates consistent improvements across diverse anomaly detection models, such as OCSVM, LSTM-VAE, CNN-GRU-Autoencoder, and CNN-LSTM-Autoencoder. Moreover, recent studies published after 2020 continue to highlight data scarcity and the necessity of synthetic data generation as critical challenges in ICS security research [6–8,10–15]. In this context, the present study retains its relevance and value by providing a practical framework that bridges the gap between limited real datasets and the data requirements of deep learning models. Ultimately, this work contributes to advancing the reliability and practicality of AI-driven anomaly detection in industrial control systems.

2. Theoretical Background

2.1. Overview of ICS

Automation and control in the industrial sector are recognized as core elements of modern industry, contributing to increased productivity and improved efficiency. Accordingly, industrial control systems (ICSs) are actively used across various industrial domains. ICSs are automated systems that control and monitor essential operations in factories, power plants, transportation facilities, and other critical infrastructure. These systems are vital in environments where stability and reliability are critical. However, an ICS also presents inherent security issues and vulnerabilities, making secure operations and data protection critical challenges.

Most ICSs, which operate and control the functions of key national infrastructure, are run within closed networks to ensure operational stability and security. While such closed networks offer a security advantage by separating internal systems from external networks—thereby limiting external access—they also bring about administrative inconveniences due to their isolated nature.

ICS refers broadly to various control mechanisms and associated measurement devices applied to industrial technologies used in fields such as product manufacturing and power generation. An ICS can generally be categorized into three functional areas: visualization, monitoring, and control.

Research on anomaly detection in ICSs has evolved from traditional statistical approaches to more advanced machine learning and deep learning techniques. Recently, there has been a growing trend toward adopting hybrid deep learning algorithms that integrate multiple deep learning models to enhance anomaly detection performance.

2.2. Anomaly Detection in the ICS Domain

As the importance of industrial control systems (ICSs) continues to grow, anomaly detection research targeting ICS operational data has been conducted using various algorithms. However, acquiring test data by directly manipulating ICS poses several challenges. For instance, in highly sensitive ICS environments such as nuclear power plants, the potential social and economic costs of system anomalies are immense. Although anomaly detection in such systems is critically important, manipulating an operational control system solely for research purposes is practically infeasible [5].

To address this, many countries have established simulation devices to generate and utilize research datasets for anomaly detection in ICSs. The iTrust research lab at the Singapore University of Technology and Design (SUTD) has built simulated test environments for power control systems, drainage systems, and water treatment systems. These testbeds have enabled the generation of attack datasets based on diverse scenarios. In particular, the water treatment simulation device models a municipal water treatment process and includes a dataset of 36 complex and varied attack types. This dataset, known as the Secure Water Treatment (SWaT) testbed, is widely used for developing and evaluating anomaly detection models in various ICS applications [15].

In the domain of industrial control systems (ICSs), research on anomaly detection has been actively conducted with a focus on dataset construction, modeling techniques, interpretability, and practical applicability. A recent study introduced the ICS-Flow dataset, which integrates both network traffic and process logs to overcome the lack of standardized benchmarks and to enable fair performance comparisons across anomaly detection algorithms [13]. Another line of work proposed a Modbus-NFA behavior model that captures the behavioral flow of the Modbus/TCP protocol using nondeterministic finite automata, thereby detecting deviations from legitimate operational patterns [14].

Beyond behavioral modeling, Autoencoder-based approaches have also been refined to detect subtle anomalies in ICS time-series data with higher precision [9], while deep learning-based intrusion detection systems have been applied to SCADA environments to achieve superior detection rates and real-time responsiveness compared to traditional classifiers [13]. Other studies provided taxonomies of cyberattacks and optimization strategies in ICS security, emphasizing the importance of multilayered defense mechanisms [14].

The issue of interpretability has also drawn increasing attention. One study systematically evaluated attribution techniques in machine learning-based ICS anomaly detection, revealing that the quality of explanations depends heavily on the timing of detection and the nature of the attack [15]. Comparative evaluations of anomaly detection models on public ICS datasets demonstrated that the best-performing model varies depending on data characteristics, and that satisfactory performance can be achieved even with limited training data [6]. Further advancements include a multi-level graph attention network that integrates process and communication graphs, thereby improving both detection accuracy and the localization of anomalies [7].

In parallel, privacy-preserving and distributed approaches have been explored. A federated learning-based framework, termed Cyber Sentry, has shown that anomaly detection performance can be maintained across organizations without centralized data sharing [14]. Traditional machine learning methods have also been investigated for intrusion detection using both process state and network traffic data, highlighting strengths and weaknesses compared to deep learning methods [16]. A GRU-based interpretable multivariate time-series anomaly detection model has been proposed to capture complex dependencies across variables while providing feature-level contributions for enhanced interpretability [17]. More recently, a distributed linear deep learning model has been developed to ensure scalability and real-time performance in resource-constrained ICS environments [18].

In summary, prior studies have addressed ICS anomaly detection from multiple perspectives, including realistic dataset construction [13], protocol- and behavior-based modeling [14], autoencoder- and deep neural network-based methods [13,15], systematic reviews of threats and defensive strategies [14], interpretability [15,17], comparative model evaluations [6], graph neural network extensions [7], federated learning for collaborative defense [8], comparative analysis of machine learning and deep learning approaches [16], and distributed lightweight model design [18]. These efforts collectively provide a foundation and motivation for the methodology proposed in this work.

Nevertheless, despite these advancements, one of the most persistent challenges emphasized in recent studies (2020 onward) remains the scarcity of high-quality labeled data for ICS anomaly detection. While many methods assume the availability of sufficient datasets, practical deployment is often hindered by this limitation. In this context, the present study contributes unique value by introducing a Semi-Supervised GAN (SGAN) framework that synthesizes realistic time-series ICS data. By augmenting limited datasets and empirically validating improvements across multiple anomaly detection models, this work directly addresses the data scarcity bottleneck and strengthens the applicability of AI-driven anomaly detection in real-world ICS environments.

2.3. Deep Learning in Anomaly Detection

In recent years, deep learning has emerged as a key approach in anomaly detection across various domains, including finance, healthcare, cybersecurity, and industrial systems. Unlike traditional machine learning methods that rely heavily on manual feature extraction, deep learning models are capable of automatically learning hierarchical feature representations from raw data, making them particularly effective for complex pattern recognition tasks such as anomaly detection [19].

Among the various deep learning architectures, models such as Autoencoders, Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), have been widely used. These models are particularly suited for time-series data, which is common in industrial control systems, as they can capture temporal dependencies and detect subtle deviations from normal patterns [20].

Furthermore, recent research has focused on combining multiple deep learning techniques into hybrid models to enhance detection performance. For instance, integrating CNNs with LSTM or GRU layers enables the model to extract both spatial and temporal features, improving its ability to detect complex anomalies. As the availability of high-quality datasets increases, the application of deep learning in anomaly detection is expected to become even more widespread and impactful.

2.4. Convolutional Neural Network Model

Convolutional Neural Networks (CNNs) are a type of deep learning model specifically designed to process data arranged in grid-like patterns, such as images [21–23]. A typical CNN architecture consists of convolutional layers, pooling layers, and fully connected layers. The convolution and pooling layers are responsible for extracting features from the input images, while the fully connected layer performs the classification task. Among these, the convolutional layer plays a central role.

In digital images, pixel values are represented in a two-dimensional grid, which can be treated as a matrix. CNNs utilize a small grid-like parameter known as a “kernel” to extract features from the image. These features, extracted via the kernel, significantly enhance the overall efficiency of the learning process [24].

The convolutional layer has two key hyperparameters: kernel size and the number of filters. This layer divides the input image data into fixed-size local patches that match the kernel size used by each filter. Each filter in the convolutional layer has the same depth as the input data. Multiple filters—each composed of different weight values—are used so that the network can learn a wide range of distinctive features. An activation function is then applied to the result, allowing for the network to extract meaningful characteristics from the original image. Ultimately, the convolutional layer produces a set of feature maps that serve as input to subsequent convolutional or pooling layers [25].

2.5. Hybrid Deep Learning Model

In the field of time-series data processing, CNNs have been explored in several studies as a solution to limitations observed in RNNs and LSTMs. In particular, difficulties arise when learning long-term patterns in sequences exceeding 100 time steps. One effective solution involves using 1D convolutional layers to reduce the input sequence length. By applying the principle of feature map extraction using kernel filters in the convolutional layer, 1D convolutional layers slide several kernels over the sequence, generating a distinct 1D convolutional layer for each kernel. Each kernel learns to detect a short sequential pattern. For instance, using ten kernels results in ten 1D sequences, which can be interpreted as a single 10-dimensional sequence. This approach allows for the construction of hybrid neural networks that combine recurrent layers with 1D convolutional or pooling layers [26].

RNN layers are inherently designed for sequential processing. To compute the output at time step t , outputs from t_0 to t_{t-1} must first be calculated, which hinders parallelization. In contrast, 1D convolutional layers do not maintain state information across time steps, enabling a parallelizable computational structure. Instead of requiring all prior outputs to compute the output at a given time step, the model can compute based on small feature maps applied to the input. Unlike recurrent layers, 1D convolutional layers are less affected by unstable gradients. Therefore, incorporating one or more 1D convolutional layers into an RNN allows for efficient preprocessing of the input, enabling the model to discard insignificant details while preserving useful information [27].

2.6. SWaT Dataset

The Secure Water Treatment (SWaT) testbed was developed at the Singapore University of Technology and Design (SUTD) as a scaled-down, fully operational water treatment plant as seen in Figure 1.

The process consists of six stages (P1–P6). P1 stores raw water in tank T101, with level monitored by LIT101 and controlled via MV101 and pump P101. P2 injects chemicals (HCl, NaOCl, NaCl) using dosing pumps (P201, P203, P205) and monitors quality via pH, chlorine, and conductivity sensors (AIT201–203). P3 feeds water from tank P301 through ultrafiltration (UF) membranes, where LIT301 and DPIT301 regulate tank level and membrane fouling. P4 supplies water to the reverse osmosis (RO) unit via tank P401, including dechlorination with UV and NaHSO₃ dosing (AIT402). P5 performs RO filtration using P501 and a cartridge filter, producing permeate stored in P501 and reject water sent to backwash. P6 manages UF backwash with tank and pump P602, using RO reject water to periodically clean UF membranes. Sensors such as LIT, FIT, AIT, and DPIT across all stages provide monitoring and control through the SCADA system.

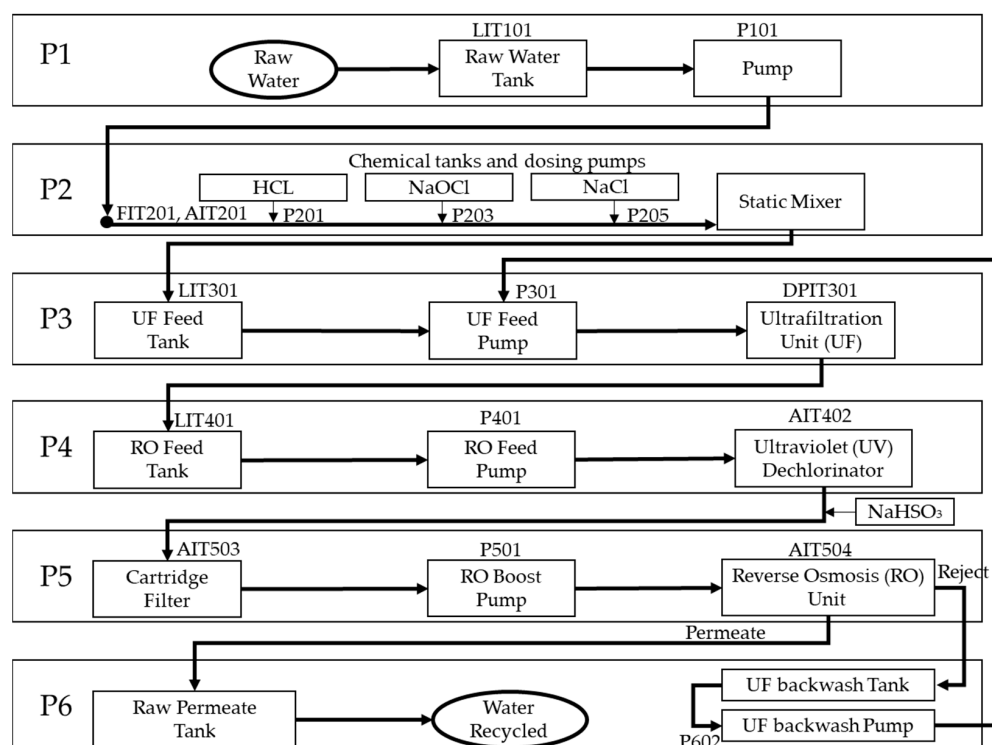


Figure 1. Block diagram of the water treatment process (recreated by the author of this paper with reference to [9]).

The dataset comprises seven days of normal system operation and four days of data during which 36 attack scenarios were executed. In total, the dataset contains 946,722 records, each labeled as either normal or attack data across 51 features derived from sensor and actuator readings. The threat model assumes that the system has already been compromised by an attacker. Attackers either issue malicious commands to the PLCs or override legitimate instructions to simulate normal system states while causing physical disruption. The SWaT dataset includes a table detailing the nature and timing of each attack. Each attack is designed to have a specific physical impact on the system. For example, Attack 30 aims to deplete the water level in the first-stage tank by fixing the LIT101 level sensor reading at 700 mm while leaving the discharge pump P101 continuously on for 20 min. The effects of each attack are assessed based on the intended disruption, the resulting system behavior, and the time required for the system to return to a stable state. All attack scenarios are predefined and labeled according to the corresponding actuator commands and system responses [15].

2.7. HAI 22.04 Dataset

The HAI (HIL-based Augmented ICS) 22.04 dataset was developed by the Korea Institute of National Security Research in 2022, based on a Hardware-In-the-Loop (HIL) ICS testbed equipped with industrial-grade control devices from GE, Emerson, and Siemens. The testbed simulates multiple industrial processes, including a steam turbine, pumped storage, boiler, and water treatment systems, each controlled by three PLCs and connected to various actuators (motors, valves, switches) and sensors (measuring pressure, temperature, and flow). Data is collected through an OPC-UA gateway, which uses the OPC Unified Architecture protocol to integrate heterogeneous devices.

The dataset consists of multivariate time-series data containing both normal operation and cyberattack scenarios. Attack data were generated from 38 scenarios targeting six control loops, simulating a variety of ICS-specific threats. The HAI 22.04 version improves

upon earlier releases (e.g., HAI 20.07) by providing richer sensor coverage, refined labeling of attack intervals, and longer continuous sequences for both normal and abnormal operations.

Because the testbed is constructed with real industrial devices and replicates complex physical processes, the dataset is considered highly realistic and has become a widely used benchmark for evaluating ICS anomaly detection algorithms [1]. However, its limitations should also be noted: while it simulates diverse processes, it cannot capture the full complexity and unpredictability of live industrial environments, and the attack scenarios remain bounded to predefined cases. Moreover, certain industry-specific operations, such as chemical or large-scale power grid processes, are not fully represented. Thus, while HAI 22.04 provides a valuable and practical proxy for real ICS operations, it must be applied with an awareness of these constraints.

2.8. SGAN (Semi-Supervised Learning GAN)

The Semi-Supervised Generative Adversarial Network (SGAN) is an extension of the standard Generative Adversarial Network (GAN) designed to address semi-supervised learning tasks. Semi-supervised learning involves training models with a small amount of labeled data combined with a large amount of unlabeled data, enabling the model to leverage additional information without requiring extensive manual labeling.

A conventional GAN consists of two neural networks—a generator and a discriminator—trained in an adversarial manner. The generator learns to produce synthetic samples that resemble real data, while the discriminator aims to distinguish between real and synthetic samples. SGAN modifies the discriminator to perform an additional task: classifying real samples into their respective categories. This dual-role discriminator allows for the model not only to differentiate between real and generated inputs but also to make accurate class predictions for labeled data.

By jointly training on labeled and unlabeled data, SGAN is able to extract richer feature representations, learn more generalized decision boundaries, and mitigate the limitations of data scarcity. The generator, in turn, is encouraged to create class-consistent synthetic samples that can improve the robustness of downstream classification models.

SGAN has demonstrated effectiveness across multiple domains, including computer vision, natural language processing, and, more recently, industrial control system (ICS) anomaly detection. For example, in image classification, large-scale unlabeled datasets can be utilized alongside a smaller labeled set, reducing annotation costs while achieving competitive or superior performance compared to fully supervised approaches. In the context of ICS, SGAN can generate synthetic operational and attack data that preserve system-specific patterns, enabling more accurate detection of anomalies even under limited labeled data conditions [28].

2.9. Limitations of Existing Methods

Despite the progress achieved in anomaly detection for industrial control systems (ICSs), existing approaches still face notable limitations when confronted with the persistent challenge of data scarcity. Traditional GAN-based frameworks have demonstrated the ability to generate synthetic data resembling real-world signals; however, their discriminators are restricted to a binary task of distinguishing between real and generated samples. As a result, conventional GANs are unable to leverage class information, thereby limiting their utility in semi-supervised settings where only a small portion of data is labeled. This shortcoming reduces their effectiveness in scenarios where labeled ICS attack data are scarce and costly to obtain.

Other generative models, such as Variational Autoencoders (VAEs) and TimeGAN, have been proposed to address data generation in time-series domains. While VAEs can produce smooth latent representations, they often suffer from blurry reconstructions and lower fidelity in capturing sharp temporal variations inherent in ICS datasets. Similarly, TimeGAN introduces temporal dynamics into the generation process, yet it requires large amounts of high-quality labeled data for stable training. Consequently, both VAE and TimeGAN exhibit limited performance when applied to the unique constraints of ICS anomaly detection, where training datasets are highly imbalanced and labeling incurs significant operational risks.

Semi-supervised learning methods outside the GAN paradigm also present drawbacks. Approaches that combine small labeled subsets with large unlabeled corpora often rely on assumptions about feature distributions that do not align with the complex temporal dependencies of ICS processes. Furthermore, their performance is highly sensitive to the proportion of labeled data, leading to reduced robustness under real-world ICS conditions.

In this context, the Semi-Supervised Generative Adversarial Network (SGAN) provides a distinct advantage. By extending the discriminator's role to both real/fake discrimination and multi-class classification, SGAN is capable of simultaneously learning from limited labeled data and exploiting the abundance of unlabeled data. This dual capability allows for the generator to produce synthetic samples that not only approximate real ICS behavior but also remain class-consistent, effectively augmenting scarce datasets. Compared with traditional GANs, VAEs, and TimeGAN, SGAN uniquely addresses the dual challenges of data scarcity and class imbalance in ICS environments, enabling improved anomaly detection performance across diverse deep learning models.

3. Research Method

The primary objective of this study is to address the challenge of anomaly detection in industrial control system (ICS) time-series data under conditions of limited training data. Specifically, the research aims to evaluate whether synthetic data generated by a Semi-Supervised Generative Adversarial Network (SGAN) can improve the performance and generalizability of deep learning models when combined with real-world ICS datasets.

To achieve this objective, the methodology was designed around three core components: (1) dataset construction and preprocessing, (2) model architecture design, and (3) evaluation and validation.

First, we examined the characteristics of ICS time-series data, which are defined by abrupt changes caused by the complex interactions of actuators, valves, and sensors. Recurrent Neural Network (RNN) approaches, traditionally applied to time-series forecasting, often fail to provide satisfactory performance under such conditions. Recent studies have introduced CNN-based approaches, originally developed for image recognition, into the time-series domain with promising results. Building on this foundation, hybrid deep learning models—particularly CNN-LSTM-Autoencoder architectures—have shown effectiveness in capturing both local patterns and long-term dependencies in time-series data.

Second, the HAI 22.04 dataset was used as the primary dataset for training, validation, and testing. The dataset was randomly split in an 80:15:5 ratio, and missing values were handled using last-value imputation. The final dataset, consisting of 911,603 records collected over a 10-day period from a closed-loop water treatment ICS simulation device, was normalized to a $[0, 1]$ range using min-max scaling. This preprocessing ensured the data was suitable for anomaly detection tasks.

Third, the proposed model was implemented using the Keras 2.10 ((François Chollet and contributors, Google LLC, Mountain View, CA, USA), focusing on the CNN-LSTM-

Autoencoder as the anomaly detection mechanism. The Autoencoder was trained solely on normal data, with anomalies detected based on reconstruction error: inputs with low reconstruction error were classified as normal, while those with high reconstruction error were classified as abnormal. This framework allows for the model to learn the operational characteristics of ICS and detect deviations effectively.

Figure 2 presents the proposed SGAN-based semi-supervised learning model for time-series anomaly detection. The HAI 22.04 dataset was divided into training, validation, and test sets, followed by a preprocessing phase. During preprocessing, missing values were handled, and the initial normal data range was verified. Feature dimensions were then normalized. A comparison was made between the fields in the training and test sets to check for any missing variables. After this initial preprocessing, feature engineering was conducted to capture the properties of the dataset, and data scaling was performed using the MinMax Scaler function of scikit-learn 1.2.2 (scikit-learn developers, Paris, France). Overall framework of the proposed SGAN-based time-series anomaly detection model. The process begins with data preprocessing of the HAI 22.04 dataset, including handling missing values, normalization, and feature engineering through variable grouping. Synthetic time-series data are then generated using a Semi-Supervised Generative Adversarial Network (SGAN), where the generator produces realistic ICS-like sequences and the discriminator simultaneously performs real/fake discrimination and class classification. The combined real and SGAN-generated data are subsequently used to train deep learning-based anomaly detection models (e.g., CNN-LSTM-Autoencoder), with performance validated through PCA/t-SNE similarity analysis and comparative evaluation against baseline models. This framework highlights the role of SGAN in mitigating data scarcity and improving anomaly detection accuracy in ICS environments.

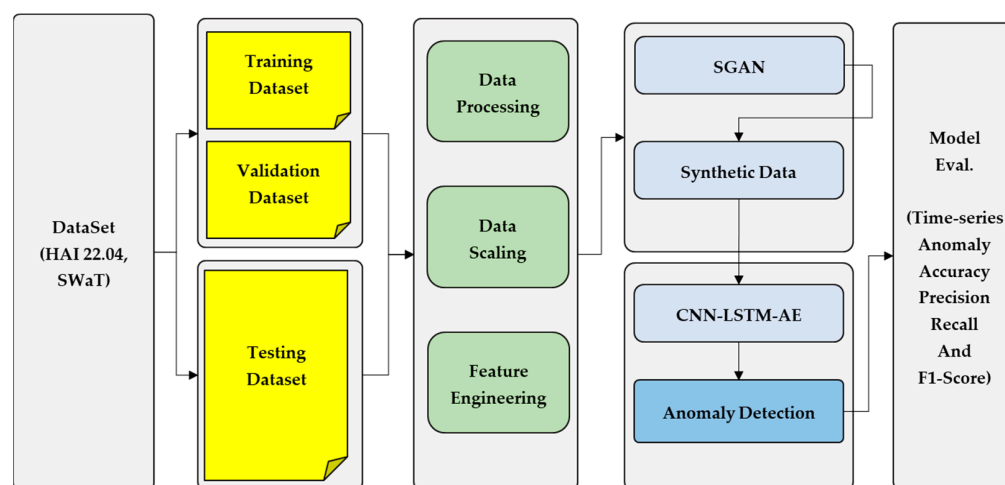


Figure 2. Time-series anomaly detection model using SGAN-based synthetic data (originally created by the authors for this study).

Min–max normalization is a technique used to normalize the range of independent variables or features in a dataset. It is commonly performed during the data preprocessing stage and is also referred to as data normalization. This technique adjusts the range of each feature to $[0, 1]$ or $[-1, 1]$, providing one of the simplest scaling methods. For a given range $[a, b]$ where a and b are the minimum and maximum values, respectively, the following formula is used for min–max normalization:

$$x_{normalised} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

3.1. Procedure

The HAI 22.04 dataset, a publicly available ICS dataset, is used as the basis for this study. Basic data preprocessing is conducted to handle missing values and outliers. Synthetic time-series datasets are generated using a Semi-Supervised Learning GAN (SGAN), and the generated data is then used as training data for a deep learning model for time-series anomaly detection. The performance of anomaly detection is compared before and after applying the semi-supervised learning model.

To evaluate model performance, a comparative analysis is conducted between models trained with SGAN-generated synthetic data and those trained without it. The evaluation aims to determine the extent to which SGAN improves the accuracy of time-series anomaly detection.

To assess the quality of the ICS time-series synthetic data generated by SGAN, similarity to real data is verified using Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE), a distance-based probabilistic statistical model.

Additionally, the SWaT dataset an ICS public dataset widely used internationally is employed for further empirical comparison. Using the same procedure, OCSVM, LSTM-VAE, CNN-GRU-AE, and the proposed CNN-LSTM-AE model are each applied to evaluate anomaly detection performance. After generating synthetic data for SWaT using SGAN, PCA, and t-SNE are used to verify the similarity to real SWaT data. Upon confirming sufficient similarity, the SGAN-generated SWaT dataset is included in the training to measure anomaly detection performance. A comparative evaluation of time-series anomaly detection performance is then conducted and discussed.

3.2. Feature Engineering and Variable Grouping

In this study, the raw dataset underwent extensive feature engineering to improve the model's ability to detect anomalies. The process began with a domain-specific analysis of the ICS architecture to understand the operational relationships between variables. Based on this analysis, variables were grouped according to their functional similarities, as summarized in Table 1.

Table 1. Grouping of ICS variables.

Variable Group	Example Variables
Pressure-related sensors	Inlet/outlet pressure indicators; tank pressure gauges
Temperature-related sensors	Heater outlet temperature; coolant temperature
Flow-related sensors	Pump discharge flow; valve flow rates
Operational state indicators	Pump status; valve open/close signals

This grouping allowed for the model to focus on correlated patterns within each subsystem, thereby enhancing the detection of subtle anomalies that might be diluted in a mixed-variable feature space. For instance, pressure and temperature readings from the same physical section of the process tend to exhibit predictable co-variation under normal operation; deviations from this correlation can be indicative of faults or attacks.

After grouping, each subset of variables underwent preprocessing tailored to its characteristics. Continuous numerical features were standardized using z-score normalization to center them around zero with unit variance, while bounded signals (e.g., valve position indicators) were min-max-scaled to [0, 1]. Categorical or binary control signals were retained in their original format but encoded appropriately for model ingestion. Outlier

detection was performed within each group to identify and replace sensor dropouts or corrupted readings, using statistical thresholds based on interquartile ranges.

Finally, the grouped and normalized features were concatenated into the model's input sequence, preserving group boundaries through feature indexing. This structured representation improved the learning efficiency of both the autoencoder and SGAN components, as the models could exploit the inherent relationships between grouped features without being distracted by unrelated noise from other subsystems.

3.3. SGAN-Based Synthetic Data Generation

The synthetic data generation method proposed in this paper utilizes a Semi-Supervised Generative Adversarial Network (SGAN) model. The basic structure of a GAN consists of a generator network and a discriminator network that operate in opposition, enabling the generator to produce synthetic data that can be classified as real by the discriminator.

Algorithm 1 illustrates the structure of the SGAN model used to generate synthetic ICS time-series datasets. First, the actual ICS time-series dataset is input, followed by data analysis. In this step, a correlation analysis is conducted to examine relationships and distributions among data dimensions. In addition, missing values and variability in the data are assessed. Based on the analysis results, dimensionality reduction is performed to construct a computationally efficient data processing environment.

Algorithm 1: Pseudocode of SGAN-based synthetic ICS time-series generation

Inputs:

X_{raw} // multivariate ICS time-series ($T \times D$)
 w, s // window length and stride
 scaler // normalization method
 Z_{dim} // latent noise dimension
 $G(\cdot; \theta_G), D(\cdot; \theta_D)$ // Generator and Discriminator
 η_G, η_D // learning rates
 B, E // batch size and epochs

Outputs:

$X_{\text{syn_full}}$ // synthetic time-series
 Report // validation results

Procedure:

1. Normalize X_{raw} using scaler and create window dataset S_{real} with (w, s) .
2. Define adversarial losses:

$$L_D = -E[\log D(S)] - E[\log(1 - D(G(z)))]$$

$$L_G = -E[\log D(G(z))]$$
3. For epoch = 1 ... E:
 - For minibatch B_{real} from S_{real} :
 - Update θ_D with L_D using real and fake samples
 - Update θ_G with L_G using generated samples
4. Generate synthetic windows $S_{\text{fake}} = \{G(z)\}$.
5. Stitch S_{fake} into long sequence:
 - if $s < w$: apply overlap-add; else concatenate.
6. Inverse transform to obtain $X_{\text{syn_full}}$.
7. Validate (PCA/t-SNE, stats, TaPR) and compile Report.
8. Save $X_{\text{syn_full}}$, model parameters, and logs.

Return $X_{\text{syn_full}}$, Report.

To describe the learning mechanism of the SGAN model in more detail, this section presents the objective functions for both the generator and the discriminator, followed by a

step-by-step training procedure. In this study, we adopt the Semi-Supervised Generative Adversarial Network (SGAN) framework, where the discriminator is extended to perform both real/fake discrimination and multi-class classification. The total loss function L_{SGAN} consists of the following components Equations (2)–(5) [27]:

The generator is trained to fool the discriminator into classifying generated data as real.

Generator Loss L_G :

$$L_G = -E_{z \sim P_z(z)} [\log D_{real}(G(z))] \quad (2)$$

Discriminator Loss L_D :

$$L_D = L_{adv} + L_{cls} \quad (3)$$

The discriminator's loss is composed of

1. Adversarial Loss L_{adv} :

$$L_{adv} = -E_{x \sim P_{data}(x)} [\log D_{real}(x)] - E_{z \sim P_z(z)} [\log (1 - D_{real}(G(z)))] \quad (4)$$

Explanation:

x : Real data from the dataset.

$D_{real}()$: Output of the discriminator for real/fake classification.

This term encourages the discriminator to correctly distinguish real data from fake data.

2. Classification Loss L_{cls} :

$$L_{cls} = -E_{(x,y) \sim D_l} \sum_{k=1}^K 1(y=k) \log D_{class}^k(x) \quad (5)$$

Explanation:

D_l : Labeled dataset (subset of real data with ground-truth class labels).

y : True label for sample x .

K : Total number of classes.

$D_{class}^k(x)$: Discriminator's predicted probability that input x belongs to class k

$1(y=k)$: Indicator function that is 1 when $y=k$, 0 otherwise.

3.4. SGAN Network Structure

To improve reproducibility and ensure clarity in methodology, the Semi-Supervised Generative Adversarial Network (SGAN) used in this study is described in detail below. Figure 3 illustrates the complete network architecture, including the generator and discriminator components.

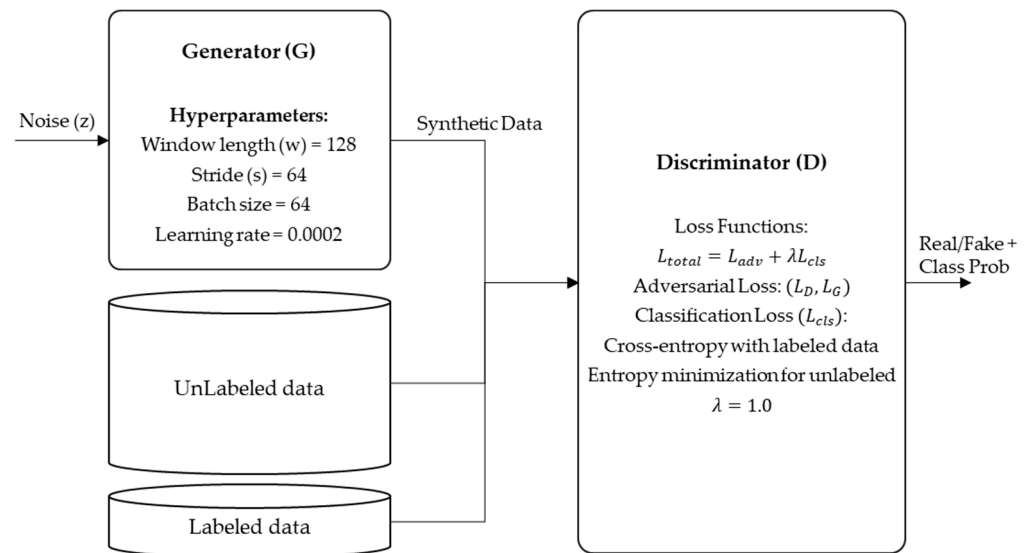


Figure 3. SGAN structure with hyperparameters and loss functions (originally created by the authors for this study).

Generator (G):

The generator receives a 100-dimensional random noise vector sampled from a standard normal distribution and outputs synthetic ICS time-series windows of dimension ($w \times d$), where w is the window length and d is the number of features (32 for the HAI 22.04 dataset). The architecture consists of four fully connected layers with output dimensions of 128, 256, 512, and ($w \times d$). Each hidden layer applies a LeakyReLU activation function ($\alpha = 0.2$), followed by Batch Normalization to stabilize training. Dropout layers with a rate of 0.3 are inserted after the second and third dense layers to prevent overfitting.

Discriminator (D):

The discriminator receives input sequences of dimension ($w \times d$). The inputs are derived from three sources: (1) a small portion of labeled data, which includes ground-truth class information (normal or attack categories); (2) a large amount of unlabeled data, which does not provide explicit class labels but allows for the discriminator to learn feature distributions and perform pseudo-labeling through entropy minimization; and (3) synthetic data generated by the generator.

The discriminator consists of three fully connected layers with output sizes of 512, 256, and ($C + 1$), where C denotes the number of labeled classes, and the additional “+1” represents the fake class. The first two layers use LeakyReLU activation ($\alpha = 0.2$) and Dropout (rate = 0.4). The final layer applies a softmax activation function to output probabilities: labeled data are classified into one of the real classes, unlabeled data are assigned probabilistic distributions, and synthetic data are classified into the fake class.

Loss Function:

The total loss function combines adversarial loss and classification loss:

$$L_{total} = L_{adv} + \lambda L_{cls} \quad (6)$$

Here, L_{adv} includes adversarial objectives for the discriminator and generator, while L_{cls} denotes the semi-supervised classification loss:

$$L_{cls} = - \sum_{i=1}^{N_l} y_i \log(p_i) \quad (7)$$

where N_l is the number of labeled samples, y_i is the ground-truth label, and p_i is the predicted probability. Both labeled and unlabeled data are used for training, with unlabeled data contributing via entropy minimization. The weight parameter λ was empirically set to 1.0 in this study.

3.5. Hyperparameter Selection and Sensitivity Analysis

The selection of window length (w) and stride (s) is crucial for capturing temporal dependencies in ICS data. Based on domain-specific prior studies and exploratory analysis of the HAI 22.04 dataset, we initially set $w = 128$ and $s = 64$, which balances temporal coverage and computational efficiency. To evaluate robustness, we performed a sensitivity analysis using $w = 64, 128$, and 256 .

Results of Sensitivity Analysis:

$w = 64$: Synthetic sequences were too short to capture long-term dependencies, resulting in lower similarity scores under PCA/t-SNE validation.

$w = 128$: Achieved the best balance between diversity and fidelity, with high similarity to real data and stable training convergence.

$w = 256$: Produced longer sequences but increased training instability and mode collapse frequency.

This comparison confirms that $w = 128, s = 64$ yields the most reliable synthetic data quality and anomaly detection performance. The findings underscore that time window selection significantly affects the fidelity of SGAN-generated data.

4. Experiments and Results

4.1. Evaluation Metrics

This study verifies the validity and reliability of the proposed model based on its classification performance in distinguishing between normal and abnormal (anomalous) states in time-series data. Accordingly, the evaluation metrics employed are classification performance metrics, which commonly include accuracy, precision, recall, F1 score (harmonic mean), and the AUC–ROC (Area Under the Curve–Receiver Operating Characteristic) curve. The following equations (Equations (8)–(11)) illustrate the metrics used for evaluating classification performance [29].

$$accuracy = \frac{TP + TN}{P + TN + FP + FN} \quad (8)$$

$$precision = \frac{TP}{TP + FP} \quad (9)$$

$$recall = \frac{TP}{TP + FN} \quad (10)$$

$$F1\ Score = 2 \times \frac{precision \times recall}{precision + recall} \quad (11)$$

In the equations above, TP (true positives) and TN (true negatives) represent the number of correctly classified positive and negative instances, respectively. FP (false positives) and FN (false negatives) indicate misclassified instances. Accuracy provides the overall correctness of the model. Precision measures the proportion of predicted positives that are truly positive, while recall assesses the ability of the model to identify all actual positive cases. The F1 score represents the harmonic mean of precision and recall and is especially useful when evaluating models under class imbalance conditions.

An analysis of the test data composition in the HAI 22.04 dataset used in this study shows 349,170 normal instances and 12,030 abnormal instances, with normal data accounting for 96.55% of the total. Therefore, if performance evaluation is conducted using accuracy,

a validation model that classifies all data as normal would still yield a high accuracy of 96.55%, leading to misleading results. To ensure the reliability of validation in cases of imbalanced or biased data, the F1 score is used.

Time-series anomaly detection involves identifying abnormal patterns or events that deviate from expected behavior within a time-series segment. As the complexity and volume of time-series data increase, developing accurate and efficient models for anomaly detection becomes critical. However, evaluating the performance of such models poses unique challenges due to the temporal nature of the data and the imbalance between normal and abnormal instances. The F1 score, which combines precision and recall, provides a balanced measure of model performance in detecting anomalies.

TaPR (Time-series Aware Precision and Recall) is a classification performance metric specifically designed for time-series anomaly detection. Since anomalies often persist over a period once triggered, traditional classification metrics may fall short of capturing this characteristic. TaPR addresses this issue by offering a complementary evaluation metric. It is composed of TaP and TaR, which correspond to traditional precision and recall, respectively [29].

4.2. Evaluation Methods

The evaluation proceeds as follows. First, a CNN-LSTM-Autoencoder model is built based on the HAI 22.04 dataset, and hyperparameters are tuned to establish the optimal model. The weight parameters of the best-performing model are then saved in an h5 file (HDF5: Hierarchical Data Format) so they can be reloaded and used for subsequent comparative validation experiments.

Next, the SGAN is used to generate a time-series synthetic dataset based on the HAI 22.04 dataset. The SGAN model's hyperparameters are adjusted to identify the weights that yield the best performance. Multiple synthetic datasets are created by adjusting the proportion of anomaly segments in the test set of the synthetic HAI 22.04 dataset. After evaluating the performance of the CNN-LSTM-Autoencoder model trained on the synthetic dataset, comparative validation is conducted against the model trained without synthetic data.

4.3. Experimental Results of the ICS Anomaly Detection Model

The testing of the anomaly detection model began with a preparation phase involving hyperparameter tuning to establish a model with optimal performance. Hyperparameters were adjusted heuristically, with epoch count and batch size serving as the main parameters.

The hyperparameters applied in the proposed time-series anomaly detection model are shown in Table 2. The same hyperparameters were used for both pre-training and fine-tuning phases. The validation data was set at a ratio of 0.2 relative to the training data. The batch size, which indicates the number of data samples used in one training iteration, was set to 128, taking into account the hardware performance of the experimental setup. The number of training iterations (epochs) was set to 50. In addition, early stopping was applied to prevent overfitting. The loss function was set to MSE (Mean Squared Error), and the optimizer was configured as Adam.

Table 2. Hyperparameters of the time-series anomaly detection model.

Hyperparameter	Pre-Training	Fine-Tuning
Validation Ratio	0.2	0.2
Batch Size	128	128
Epoch	50	50
Optimizer	Adam	Adam
Loss Function	MSE	MSE

The model for generating ICS synthetic datasets based on an SGAN (semi-supervised learning GAN) consists of a generator that creates fake HAI 22.04 datasets from random vector inputs and a discriminator that distinguishes the generated data as synthetic datasets highly similar to real HAI 22.04 data.

The SGAN model proposed in this study was implemented using the Keras 2.10 (François Chollet and contributors, Google LLC, Mountain View, CA, USA) with a TensorFlow backend in Python 3.10 (Python Software Foundation, Wilmington, DE, USA). The generator consists of four dense layers with output sizes of 128, 256, 512, and the target data dimension, respectively. Each dense layer uses the LeakyReLU activation function ($\alpha = 0.2$), followed by Batch Normalization to stabilize training and improve convergence. A Dropout layer (rate = 0.3) is applied after the second and third dense layers to prevent overfitting.

The discriminator is composed of three dense layers with output sizes of 512, 256, and the number of output classes + 1 (for the “fake” class). The first two layers employ LeakyReLU activation ($\alpha = 0.2$), while the output layer uses a softmax function to output class probabilities. The discriminator also incorporates a Dropout rate of 0.4 to improve generalization. The input dimension for both generator and discriminator matches the feature vector size of the preprocessed dataset (HAI 22.04 dataset in this case). The generator input is a random noise vector of dimension 100, sampled from a standard normal distribution. The SGAN was trained using the Adam optimizer with a learning rate of 0.0002 and $\beta_1 = 0.5$ for both generator and discriminator. The batch size was set to 64, and training proceeded for a maximum of 200 epochs with early stopping (patience = 20) based on validation loss to prevent overfitting.

All experiments were conducted on a workstation equipped with an Intel Core i9-12900K CPU (Intel Corporation, Santa Clara, CA, USA), 64 GB RAM (Samsung Electronics, Suwon, Republic of Korea), and an NVIDIA RTX 3090 GPU (NVIDIA Corporation, Santa Clara, CA, USA), running Ubuntu 22.04 LTS. Python 3.10 (Python Software Foundation, Wilmington, DE, USA), Keras 2.10 (François Chollet and contributors, Google LLC, Mountain View, CA, USA), and TensorFlow 2.11 (Google LLC, Mountain View, CA, USA) were used for model development and training. The dataset was split into training and testing sets using an 80:20 ratio, with the training set further divided into training and validation subsets (90:10 split). The early stopping criterion was applied using the validation subset to ensure optimal generalization without unnecessary training epochs.

Figure 4 illustrates the distributional similarity between the synthetic and original datasets, visualized through PCA and t-SNE dimensionality reduction techniques, with PCA on the left and t-SNE on the right in the six representative sample groups comparing real data (blue) with SGAN-generated synthetic data (orange). To ensure robustness, the evaluation was conducted by splitting the dataset into test groups under a five-fold cross-validation scheme, and the visualizations shown here represent aggregated results across these folds.

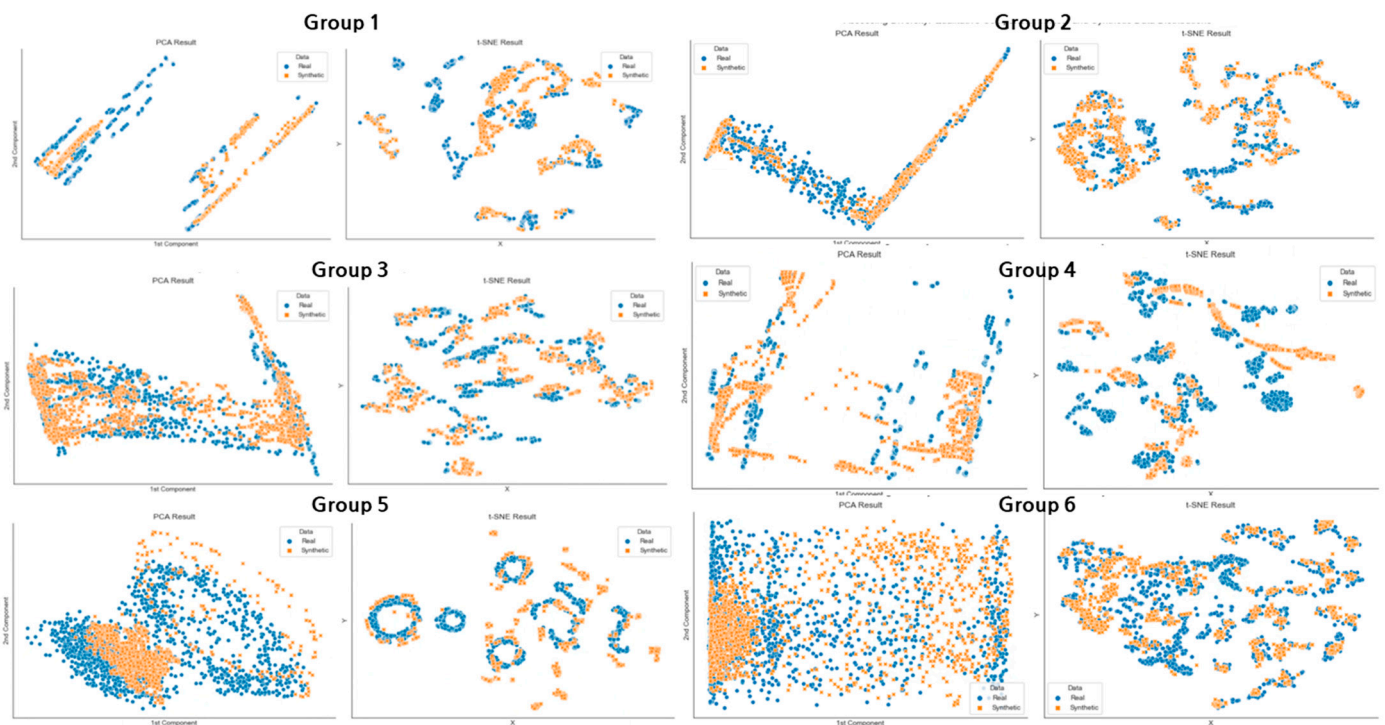


Figure 4. PCA and t-SNE analyses of six representative sample data groups.

The quality of the time-series synthetic data generated with SGAN was assessed using two criteria. First, to evaluate diversity, the real and synthetic HAI 22.04 datasets—each with 128-time steps and 32 features—were visualized on a two-dimensional graph. For quantitative diversity assessment, Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) were applied.

The execution results display the progress of the t-SNE algorithm. For each sample in t-SNE data, the 121 nearest neighbors were identified, after which conditional probabilities were computed, representing the likelihood that two samples are considered neighbors in the low-dimensional space. The average variance of the Gaussian kernel used to calculate these probabilities was approximately 0.320242.

Table 3 summarizes the performance comparison between the proposed model and a wide range of baseline models when trained on the SGAN-generated synthetic dataset. On the HAI 22.04 dataset, the proposed SGAN + CNN-LSTM-AE model achieved a precision (TaP) of 99.81, recall (TaR) of 91.30, and an F1 score of 95.42. In comparison, the CNN-LSTM-AE model trained without SGAN synthetic data recorded a precision of 99.90, recall of 86.92, and an F1 score of 93.03, confirming an improvement of 2.39 F1 points when incorporating synthetic data.

When compared to other baselines, including traditional anomaly detection models (OCSVM, LSTM-VAE), deep learning-based Autoencoder variants (CNN-GRU-AE, CNN-LSTM-AE), and more recent generative models (TimeGAN, RCGAN), the proposed SGAN + CNN-LSTM-AE consistently outperformed all alternatives across both datasets. Notably, on the SWaT dataset, the proposed model reached a precision of 99.73, recall of 97.06, and an F1 score of 97.50, surpassing the performance of all generative and semi-supervised baselines.

These results highlight three important findings:

- (1) The integration of SGAN significantly enhances anomaly detection performance, particularly by improving recall, which demonstrates a stronger capability for identifying anomalous patterns.

- (2) The consistently high precision values indicate that the SGAN-augmented model reduces false positives, an important factor in ICS environments where operational safety is critical.
- (3) Compared with state-of-the-art generative approaches such as TimeGAN and RCGAN, the proposed SGAN demonstrates superior robustness under limited labeled data conditions, confirming the unique advantage of the semi-supervised framework.

Furthermore, statistical significance testing based on five-fold cross-validation and paired *t*-tests ($p < 0.05$) confirmed that the observed improvements are not due to random fluctuations but represent consistent and reliable gains in model performance.

Table 3. Performance comparison of anomaly detection models based on SGAN synthetic data.

Model	HAI 22.04			SWaT		
	TaP	TaR	F1 Score	TaP	TaR	F1 Score
OCSVM	79.22	44.51	56.04	45.34	59.07	56.30
LSTM-VAE	49.92	98.87	66.01	75.02	89.47	82.23
CNN-GRU-AE	97.35	85.32	90.61	95.90	90.30	92.90
CNN-LSTM-AE	99.90	86.92	93.03	97.33	92.81	94.32
TimeGAN	92.48	81.25	91.84	90.03	87.34	91.40
RCGAN	90.13	85.77	90.37	85.62	83.71	84.02
SGAN + CNN-LSTM-AE (Proposed Model)	99.81	91.30	95.42	99.73	97.06	97.50

5. Conclusions

This study addressed the challenge of data scarcity in Industrial Control Systems (ICS) anomaly detection by proposing a Semi-Supervised Generative Adversarial Network (SGAN) framework for synthetic time-series data augmentation, combined with deep learning-based models such as the CNN-LSTM-Autoencoder. Using the HAI 22.04 dataset and validated with the SWaT benchmark, our experiments confirmed that models trained with both real and SGAN-generated data achieved superior anomaly detection performance compared to those trained with real data alone.

The experimental outcomes provide robust quantitative evidence supporting the effectiveness of our approach. Specifically, the proposed SGAN + CNN-LSTM-AE model consistently outperformed the baseline CNN-LSTM-AE across both the HAI 22.04 and SWaT datasets, with improvements of 2.39 and 3.18 F1 score points, respectively. These gains were confirmed to be statistically significant ($p < 0.05$), reinforcing the conclusion that the performance improvements are attributable to the SGAN-based data augmentation rather than random chance. Such consistent results highlight the practical value of our method for enhancing anomaly detection in ICS environments.

Although hardware constraints require variable grouping and restricted full multi-variate modeling, the results clearly demonstrate that generative models can effectively enhance anomaly detection in environments where real data are limited. This contribution highlights the value of SGAN-based synthetic data not only for ICSs but also for other domains facing similar challenges in data acquisition, including healthcare, defense, and cybersecurity.

The main contribution of this work lies in showing that SGAN-generated synthetic data can serve as a practical and generalizable solution to strengthen the robustness and performance of anomaly detection models. Future research will focus on extending the approach to larger-scale and more diverse datasets, addressing anomaly scarcity in imbal-

anced data, and further improving the fidelity and efficiency of synthetic time-series data generation. In addition, integration with real-time ICS testbeds and applications in predictive maintenance and manufacturing environments such as semiconductors, automotive, and food production will be explored.

Author Contributions: Conceptualization, G.G.; Formal analysis, C.H.; Investigation, G.G.; Writing—original draft, C.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shin, H.; Lee, W.; Yun, J.; Kim, H. HAI 1.0: HIL-based augmented ICS security dataset. In Proceedings of the 13Th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20), Boston, MA, USA, 10 August 2020. <https://www.usenix.org/Availableonlineg/conference/cset20/presentation/shin> (accessed on 7 September 2025).
2. Choi, W.; Kim, J. Unsupervised learning approach for anomaly detection in industrial control systems. *Appl. Syst. Innov.* **2024**, *7*, 18. [\[CrossRef\]](#)
3. Hoh, M.; Schöttl, A.; Schaub, H.; Wenninger, F. A generative model for anomaly detection in time series data. *Procedia Comput. Sci.* **2022**, *200*, 629–637. [\[CrossRef\]](#)
4. Sundaram, S.; Hulkund, N. Gan-based data augmentation for chest x-ray classification. *arXiv* **2021**, arXiv:2107.02970. [\[CrossRef\]](#)
5. Jones, A.; Kong, Z.; Belta, C. Anomaly detection in cyber-physical systems: A formal methods approach. In Proceedings of the 53rd IEEE Conference on Decision and Control, Los Angeles, CA, USA, 15–17 December 2014; pp. 848–853. [\[CrossRef\]](#)
6. Kim, B.; Alawami, M.A.; Kim, E.; Oh, S.; Park, J.; Kim, H. A comparative study of time series anomaly detection models for industrial control systems. *Sensors* **2023**, *23*, 1310. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Lin, L.; Gu, A.; Min, F.; Zhou, S. Multi-level graph attention network-based anomaly detection in industrial control system. *Actuators* **2025**, *14*, 210. [\[CrossRef\]](#)
8. Rashid, M.H.; Naeem, A.; Aslam, N.; Fuzail, M.; Mazhar, F.; Umar, M. Cyber sentry: Strengthening security infrastructure for industrial cyber-physical systems using federated deep learning. *J. Comput. Biomed. Inform.* **2025**, *9*. Available online: <https://jcibi.org/index.php/Main/article/view/974> (accessed on 7 September 2025).
9. Goh, J.; Adepu, S.; Junejo, K.N.; Mathur, A. A dataset to support research in the design of secure water treatment systems. In *Critical Information Infrastructures Security, Proceedings of the 11th International Conference, CRITIS 2016, Paris, France, 10–12 October 2016*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 88–99. [\[CrossRef\]](#)
10. Dehlaghi-Ghadim, A.; Moghadam, M.H.; Balador, A.; Hansson, H. Anomaly detection dataset for industrial control systems. *IEEE Access* **2023**, *11*, 107982–107996. [\[CrossRef\]](#)
11. Amer, E.; Al-rimy, B.A.S.; El-Sappagh, S. Strengthening ICS defense: Modbus-NFA behavior model for enhanced anomaly detection. *J. Inf. Secur. Appl.* **2025**, *89*, 103990. [\[CrossRef\]](#)
12. Aslam, M.M.; Tufail, A.; De Silva, L.C.; Haji Mohd Apong, R.A.A.; Namoun, A. An improved autoencoder-based approach for anomaly detection in industrial control systems. *Syst. Sci. Control. Eng.* **2024**, *12*, 2334303. [\[CrossRef\]](#)
13. Diaba, S.Y.; Anafo, T.; Tetteh, L.A.; Oyibo, M.A.; Alola, A.A.; Shafie-Khah, M.; Elmusrati, M. SCADA securing system using deep learning to prevent cyber infiltration. *Neural Netw.* **2023**, *165*, 321–332. [\[CrossRef\]](#)
14. Emake, E.D.; Adeyanju, I.A.; Uzedhe, G.O. Industrial control systems (ics): Cyber attacks & security optimization. *Int. J. Comput. Eng. Inf. Technol.* **2020**, *12*, 31–41. [\[CrossRef\]](#)
15. Fung, C.; Zeng, E.; Bauer, L. Attributions for ML-based ICS anomaly detection: From theory to practice. In Proceedings of the Network and Distributed System Security (NDSS) Symposium 2024, San Diego, CA, USA, 26 February–1 March 2024. Available online: <https://www.ericwzeng.com/papers/Fung-NDSS24-ICSAtributions.pdf> (accessed on 7 September 2025).
16. Sokolov, A.N.; Pyatnitsky, I.A.; Alabugin, S.K. Applying methods of machine learning in the task of intrusion detection based on the analysis of industrial process state and ICS networking. *FME Trans.* **2019**, *47*. [\[CrossRef\]](#)
17. Tang, C.; Xu, L.; Yang, B.; Tang, Y.; Zhao, D. GRU-based interpretable multivariate time series anomaly detection in industrial control system. *Comput. Secur.* **2023**, *127*, 103094. [\[CrossRef\]](#)
18. Tang, S.; Ding, Y.; Wang, H. Industrial control anomaly detection based on distributed linear deep learning. *Comput. Mater. Contin.* **2025**, *82*, 1129. [\[CrossRef\]](#)

19. Mishra, V.K.; Palleti, V.R.; Mathur, A. A modeling framework for critical infrastructure and its application in detecting cyber-attacks on a water distribution system. *Int. J. Crit. Infrastruct. Prot.* **2019**, *26*, 100298. [CrossRef]
20. Hinton, G.E.; Osindero, S.; Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef] [PubMed]
21. Traore, B.B.; Kamsu-Foguem, B.; Tangara, F. Data mining techniques on satellite images for discovery of risk ar-eas. *Expert Syst. Appl.* **2017**, *72*, 443–456. [CrossRef]
22. Traore, B.B.; Kamsu-Foguem, B.; Tangara, F.; Tiako, P. Software services for supporting remote crisis management. *Sustain. Cities Soc.* **2018**, *39*, 814–927. [CrossRef]
23. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
24. LeCun, Y.; Boser, B.E.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.E.; Jackel, L.D. Handwritten digit recognition with a back-propagation network. *Adv. Neural Inf. Process. Syst.* **1990**, 396–404. Available online: <https://dl.acm.org/doi/abs/10.5555/5/2969830.2969879> (accessed on 7 September 2025).
25. LeCun, Y.; Cortes, C.; Burges, C.J. MNIST Handwritten Digit Database. ATT Labs Online. 2010. Available online: <http://yann.lecun.com/exdb/mnist/> (accessed on 7 September 2025).
26. Geron, A. *Processing Sequences Using RNNs and CNNs*, in *Hands on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed.; O'Reilly Medi, Inc.: Sebastopol, CA, USA, 2019; ISBN 1098122461, 9781098122461.
27. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef] [PubMed]
28. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. *Adv. Neural Inf. Process. Syst.* **2016**, *29*. Available online: https://proceedings.neurips.cc/paper_files/paper/2016/hash/8a3363abe792db2d8761d6403605aeb7-Abstract.html (accessed on 7 September 2025).
29. Hwang, W.; Yun, J.; Kim, J.; Kim, H.C. Time-series aware precision and recall for anomaly detection: Considering variety of detection result and addressing ambiguous labeling. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2241–2244. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.