future internet

Article

An Optimized Transformer–GAN–AE for Intrusion Detection in Edge and IIoT Systems: Experimental Insights from WUSTL-IIoT-2021, EdgeIIoTset, and TON_IoT Datasets

Ahmad Salehiyan [1], Pardis Sadatian Moghaddam [2] and Masoud Kaveh [3,*]

[1] School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK 74078, USA; ahmad.salehiyan@okstate.edu
[2] Department of Computer Science, Georgia State University, Atlanta, GA 30302, USA; pardis.sadatian@gmail.com
[3] Department of Information and Communication Engineering, Aalto University, 02150 Espoo, Finland
* Correspondence: masoud.kaveh@aalto.fi

Abstract

The rapid expansion of Edge and Industrial Internet of Things (IIoT) systems has intensified the risk and complexity of cyberattacks. Detecting advanced intrusions in these heterogeneous and high-dimensional environments remains challenging. As the IIoT becomes integral to critical infrastructure, ensuring security is crucial to prevent disruptions and data breaches. Traditional IDS approaches often fall short against evolving threats, highlighting the need for intelligent and adaptive solutions. While deep learning (DL) offers strong capabilities for pattern recognition, single-model architectures often lack robustness. Thus, hybrid and optimized DL models are increasingly necessary to improve detection performance and address data imbalance and noise. In this study, we propose an optimized hybrid DL framework that combines a transformer, generative adversarial network (GAN), and autoencoder (AE) components, referred to as Transformer–GAN–AE, for robust intrusion detection in Edge and IIoT environments. To enhance the training and convergence of the GAN component, we integrate an improved chimp optimization algorithm (IChOA) for hyperparameter tuning and feature refinement. The proposed method is evaluated using three recent and comprehensive benchmark datasets, WUSTL-IIoT-2021, EdgeIIoTset, and TON_IoT, widely recognized as standard testbeds for IIoT intrusion detection research. Extensive experiments are conducted to assess the model's performance compared to several state-of-the-art techniques, including standard GAN, convolutional neural network (CNN), deep belief network (DBN), time-series transformer (TST), bidirectional encoder representations from transformers (BERT), and extreme gradient boosting (XGBoost). Evaluation metrics include accuracy, recall, AUC, and run time. Results demonstrate that the proposed Transformer–GAN–AE framework outperforms all baseline methods, achieving a best accuracy of 98.92%, along with superior recall and AUC values. The integration of IChOA enhances GAN stability and accelerates training by optimizing hyperparameters. Together with the transformer for temporal feature extraction and the AE for denoising, the hybrid architecture effectively addresses complex, imbalanced intrusion data. The proposed optimized Transformer–GAN–AE model demonstrates high accuracy and robustness, offering a scalable solution for real-world Edge and IIoT intrusion detection.

Keywords: Industrial Internet of Things; intrusion detection; transformer; generative adversarial network; autoencoder; chimp optimization algorithm

check for updates

Academic Editors: Olusola Tolulope Odeyomi and Temitayo Olowu

Received: 27 May 2025
Revised: 22 June 2025
Accepted: 23 June 2025
Published: 24 June 2025

Citation: Salehiyan, A.; Moghaddam, P.S.; Kaveh, M. An Optimized Transformer–GAN–AE for Intrusion Detection in Edge and IIoT Systems: Experimental Insights from WUSTL-IIoT-2021, EdgeIIoTset, and TON_IoT Datasets. *Future Internet* **2025**, *17*, 279. https://doi.org/10.3390/fi17070279

1. Introduction

The Industrial Internet of Things (IIoT) plays a foundational role in Industry 4.0 and smart manufacturing, with applications spanning energy grids, smart factories, autonomous vehicles, oil and gas, and critical infrastructure monitoring. According to recent forecasts, the global IIoT market is expected to surpass $1.1 trillion by 2028, driven by the exponential growth in Edge computing, artificial intelligence, and 5G connectivity [1–3]. This rapid expansion brings substantial benefits in terms of operational efficiency, cost reduction, and system resilience. For instance, predictive maintenance enabled by IIoT can reduce equipment downtime by up to 30% and extend asset lifespans by over 40% [4]. Similarly, real-time anomaly detection in industrial control systems (ICS) can prevent millions of dollars in losses due to safety breaches or cyberattacks [5]. National roadmaps, such as Germany's Industrie 4.0 or the European Union's Horizon Europe initiatives, emphasize the strategic importance of IIoT in digital transformation, while also highlighting the growing cybersecurity concerns arising from the massive scale, heterogeneity, and connectivity of IIoT environments [6]. As these systems become increasingly critical to national economies and infrastructures, ensuring their security and robustness against intrusions is not only a technical challenge but also a strategic imperative [7–10].

Despite the benefits of IIoT adoption, security remains one of its most pressing and multifaceted challenges. IIoT systems operate across diverse layers, from hardware to the cloud, each with its own set of vulnerabilities [11–13]. At the hardware level, secure device identification and tamper resistance are often achieved using physical unclonable functions (PUFs), which provide lightweight, silicon-based cryptographic primitives resistant to cloning and reverse engineering [14–16]. On the network layer, the massive scale and heterogeneous nature of IIoT networks expose them to routing attacks, man-in-the-middle interception, and denial-of-service (DoS) exploits [17]. Protocols such as MQTT and CoAP, while efficient, often lack built-in security features, making them susceptible to spoofing and injection. The application layer faces risks from unauthorized access, malware injection, and insecure firmware updates, particularly in cloud-connected platforms managing industrial workloads [18]. Moreover, at the physical layer, emerging threats targeting the radio spectrum, such as eavesdropping or jamming, demand advanced defense mechanisms such as physical layer security (PLS) to ensure confidentiality through techniques such as cooperative jamming, artificial noise, and secrecy rate optimization [19–22]. The security of IIoT systems is further complicated by constrained device resources, real-time performance requirements, and diverse deployment scenarios, all of which make conventional IT-centric security solutions inadequate [23].

To mitigate security breaches and maintain trustworthiness in IIoT ecosystems, intrusion detection systems (IDS) have emerged as a vital defense mechanism [24–26]. IDSs monitor system behavior to detect abnormal activities that may indicate the presence of an attacker. Traditional IDS methods fall into three broad categories: signature-based, which compare incoming traffic against known threat patterns; anomaly-based, which flag deviations from normal behavior using statistical or machine learning techniques; and specification-based, which rely on manually defined behavior rules [27]. While signature-based approaches offer high precision against known attacks, they fail against zero-day exploits. Anomaly-based models, especially those leveraging deep learning (DL), are better suited to evolving threats but often struggle with false positives and imbalanced datasets [28]. In the context of IIoT, intrusion detection becomes substantially more complex due to the distributed and heterogeneous nature of industrial networks, low-latency demands, and the need for real-time threat detection at the edge. Moreover, IIoT-generated data are often high-dimensional, noisy, and asynchronous, requiring IDS models that can scale effectively while maintaining accuracy [29]. Lightweight IDS solutions for edge devices must also balance detection performance with computational constraints. Recent research has focused on integrating deep hybrid models, generative architectures, and temporal learning to better capture IIoT-specific attack patterns and improve detection robustness [30]. Nevertheless, ensuring efficient, scalable, and adaptive IDS in IIoT environments remains an active and critical research frontier.

1.1. Related Works

A significant body of research has been dedicated to advancing intrusion detection within IoT ecosystems using various machine learning and optimization-based frameworks. With the exponential growth of IoT devices and the escalating complexity of cyberattacks, the focus has shifted towards building more intelligent and adaptive IDS models that can handle both known and novel intrusions. Ismail et al. [10] explored how classical machine learning models can be effectively adapted for intrusion detection in IoT and IIoT scenarios. By benchmarking a variety of classifiers, including decision trees, random forests, and ensemble-based methods across the TON_IoT, WUSTL-IIoT-2021, and EdgeIIoTset datasets, they identified lightweight models best suited for deployment in environments with constrained resources. Their evaluation addressed real-time metrics such as computational efficiency and feature selection via mutual information, while also analyzing the impact of data imbalance. Notably, their transfer learning setup revealed promising generalization across different datasets, reinforcing the importance of interoperability in IDS deployment.

Kumar et al. [31] presented IoTForge Pro, a testbed specifically designed for generating high-quality intrusion detection datasets tailored to IIoT systems. The Forge IIOT dataset, created through this platform, includes diverse attack scenarios that mirror real-world IIoT environments. Their study demonstrated how various machine learning algorithms perform in classifying and detecting these threats, emphasizing the importance of custom-built datasets for evaluating IDS models. The paper contributes a valuable foundation for developing IDSs that are both robust and representative of industrial-scale traffic. Martins et al. [32] conducted a comparative evaluation of datasets used in the cybersecurity of industrial control systems, highlighting critical limitations such as narrow attack coverage and lack of support for multi-stage attacks. Their analysis revealed that while many datasets offer partial representations of threat landscapes, X-IIoTID stands out as one of the most complete resources currently available. Their findings underscore the necessity of dataset quality and completeness in the development of IDSs tailored for operational technology networks.

Ruiz–Villafranca et al. [33] proposed WFE-Tab, a novel ensemble-based model built upon the TabPFN foundation to enhance intrusion detection in IIoT systems equipped with edge computing. Recognizing the limitations of TabPFN in handling complex class structures and limited training data, they introduced a weighted fusion strategy that improves generalization across diverse attack types. The WFE-Tab model achieved high performance on the Edge-IIoTset dataset, outperforming existing benchmarks with an F1-score close to 99.81%, making it a compelling candidate for deployment in IIoT–MEC environments. In a separate contribution, Ruiz–Villafranca et al. [34] evaluated the application of TabPFN models in IIoT environments characterized by limited data availability. Their IDS approach leverages TabPFN's efficiency in low-sample settings to detect a range of attack types, benchmarking it against established models such as XGBoost and LightGBM. The study demonstrated notable performance improvements with minimal data, achieving strong F1 scores in scenarios with as few as 1000 samples. Their work positions TabPFN as a suitable choice for edge-deployable IDSs in data-sparse IIoT systems.

Hassini et al. [35] focused on building a streamlined end-to-end IDS capable of addressing the specific challenges found in industrial IoT networks. Using the Edge-IIoTset dataset, their CNN1D-based model was trained to detect a comprehensive set of 15 attack types while maintaining low complexity. Their system demonstrated impressive classification performance, with near-perfect accuracy and low loss across cross-validation folds. This work exemplifies how focused architectural design and well-chosen datasets can lead to practical and scalable security solutions for real-world IIoT deployments. Alzubi et al. [36] introduced a hybrid approach combining salp swarm optimization with a neural network-based IDS. This method enhances the learning process of multilayer perceptrons by optimizing their structure and feature selection using swarm intelligence. Evaluated across multiple benchmarks, Edge-IIoTset, WUSTL-IIOT-2021, and IoTID20, their approach delivered notable accuracy improvements compared to baseline models, demonstrating robustness across different intrusion scenarios and datasets.

Abou–Elasaad et al. [37] tackled the growing threat of cyberattacks in IIoT-powered smart grids by proposing two artificial intelligence-based IDS frameworks. Their first model, built on classical machine learning classifiers, and the second, utilizing deep learning strategies, both delivered impressive detection rates above 99%. Their systems were rigorously tested on multiple datasets, showcasing low false positive rates and high classification accuracy, thereby validating the potential of AI-driven approaches for securing mission-critical IIoT infrastructures. Singh et al. [38] addressed data privacy concerns in intrusion detection by proposing a federated learning-based IDS tailored to next-generation networks. Their design handles non-uniform data distributions, one of the primary challenges in federated settings, and introduces a class imbalance mitigation strategy that operates on both local and global model updates. Through evaluations on both IID and non-IID scenarios, the study demonstrated that the proposed Fed-IDS significantly enhances generalization and attack detection accuracy across decentralized IIoT networks. Koppula and LM [39] presented LNKDSEA, a hybrid ensemble framework combining traditional classifiers such as logistic regression, naïve Bayes, and support vector machines for IoT/IIoT attack detection. Tested on the Edge-IIoTset dataset, their model was evaluated in both binary and multi-class setups, achieving strong results in identifying a broad spectrum of attack types. The study highlights the viability of ensemble learning for lightweight yet accurate intrusion detection in constrained IoT networks.

1.2. Paper Motivation, Contribution, and Organization

As edge and industrial IoT systems grow in complexity and ubiquity, the attack surface continues to expand, making these networks prime targets for increasingly sophisticated cyber threats. Traditional intrusion detection methods often rely on static signatures or shallow heuristics, which fall short in capturing the dynamic and evolving patterns characteristic of real-world attack scenarios. Moreover, high-dimensional, imbalanced, and noisy IIoT data pose significant challenges to standalone deep learning models, resulting in unstable training, poor generalization, and high false-positive rates. There is a critical need for intrusion detection frameworks that can balance robustness, adaptability, and scalability, especially in edge environments where computational resources are limited and real-time responses are essential. This study is motivated by the convergence of three complementary capabilities: the temporal learning strength of transformers, the synthetic data generation capacity of generative adversarial networks, and the denoising and anomaly sensitivity of autoencoders. When combined within a unified architecture and enhanced through metaheuristic optimization, these components offer a promising path toward resilient, adaptive, and resource-aware intrusion detection for IIoT. We aim to address the limitations of current models by proposing a hybrid, optimized framework that learns both contextual dependencies and data distributions, while being capable of self-tuning for optimal performance across heterogeneous datasets and evolving attack vectors. The primary contributions of this paper are as follows:

- We propose a novel Transformer–GAN–AE hybrid framework for intrusion detection in Edge and IIoT environments, combining the strengths of three complementary deep learning components.
- We integrate an enhanced Chimp Optimization Algorithm (IChOA) that introduces a fifth elite agent (Balancer) for adaptive hyperparameter tuning across all model modules, improving convergence stability and model generalization.
- We utilize three diverse and comprehensive datasets, WUSTL-IIoT-2021, EdgeIIoTset, and TON_IoT, to evaluate our model's robustness and cross-domain performance under both binary and multiclass classification scenarios.
- Our architecture is rigorously benchmarked against six state-of-the-art baselines, including CNN, DBN, GAN, TST, BERT, and XGBoost, demonstrating consistent outperformance across accuracy, recall, and AUC metrics.
- We conduct detailed ablation and convergence analysis to demonstrate the individual and synergistic contributions of the Transformer, GAN, and AE modules, as well as the impact of optimization via IChOA.

The remainder of the paper is structured as follows: Section 2 details the methodology, covering the design and integration of the Transformer, GAN, AE, and IChOA modules. Section 3 outlines the experimental setup and presents the evaluation metrics and baseline models used for comparison. Section 4 discusses the results, including performance benchmarks, ablation studies, and convergence behavior. Finally, Section 5 concludes the paper and outlines future research directions.

2. Materials and Methods

This section provides a comprehensive overview of the materials and methodologies employed in designing and evaluating the proposed intrusion detection framework. We begin by introducing the datasets used in this study, including WUSTL-IIoT-2021, Edge-IIoTset, and TON_IoT, each of which offers diverse attack scenarios and data modalities relevant to edge and IIoT security. Following this, we describe the core components of our model. In Section 2.2, we explain the structure and functionality of the GAN used for learning representative distributions and augmenting minority classes. Section 2.3 focuses on the AE, which performs noise reduction and feature refinement through reconstruction-based anomaly detection. In Section 2.4, we detail the transformer encoder module, which captures long-range temporal dependencies and contextual information from sequential data using multi-head self-attention. Section 2.5 introduces the IChOA, which is employed to fine-tune hyperparameters across all model components, ensuring optimal configuration and convergence. Finally, Section 2.6 presents the complete architecture of the optimized Transformer–GAN–AE framework. This subsection describes how the individual modules are integrated into a unified pipeline, highlights the interaction between components, and illustrates the overall flow of data through the system.

2.1. Dataset

In this study, we employ three publicly available and widely used benchmark datasets to evaluate the effectiveness of our proposed Transformer–GAN–AE framework for intrusion detection in edge and IIoT environments. These datasets include WUSTL-IIoT-2021, Edge-IIoTset, and TON_IoT, each offering diverse and realistic scenarios of benign and malicious activity across various layers of IIoT systems. The datasets differ in data types, feature dimensionality, attack diversity, and sample distribution, thereby providing a comprehensive testbed for assessing our hybrid deep learning model's robustness, adaptability, and generalization capability [10].

The WUSTL-IIoT-2021 dataset was developed by Washington University in St. Louis to support the evaluation of intrusion detection systems in IIoT environments. It was collected over a testbed replicating industrial control systems (ICS) and consists of realistic IIoT network traffic under both benign and malicious conditions. The dataset emphasizes the need to model attacks that are stealthy, protocol-specific, and capable of targeting critical infrastructure. The dataset contains approximately 1,194,464 labeled instances in its raw form, with six main classes: normal, DoS, backdoor, reconnaissance, and command injection. Each instance includes 42 extracted statistical and temporal features from network traffic flows. These features capture both packet-level and flow-level behaviors, such as the number of transmitted packets, bytes sent and received, and time-based characteristics such as flow durations and inter-arrival times. The data is provided in CSV format and is fully labeled for supervised learning, with timestamps that enable sequence modeling. It is particularly suitable for evaluating models under imbalanced class conditions, as the normal class dominates the distribution. The dataset's fidelity to real-world IIoT scenarios makes it a strong benchmark for anomaly-based and behavior-driven detection frameworks, especially when paired with temporal models such as transformers.

The Edge-IIoTset dataset is a large-scale, multi-layered cybersecurity dataset specifically designed for edge-enabled and federated IoT/IIoT security research. It was collected from a sophisticated seven-layer testbed that includes cloud computing, blockchain, software-defined networking (SDN), fog and edge computing layers, and diverse IoT and IIoT devices. The dataset simulates complex attack chains across multiple network and application layers, making it one of the most comprehensive resources available for intelligent threat detection. Edge-IIoTset includes over 20 million labeled samples. It features 14 distinct attack types grouped into five threat categories: DoS/DDoS, injection-based, man-in-the-middle (MITM), information gathering, and malware. After applying feature selection, 61 high-correlation features were selected from an original set of over 1100 features. These features span system logs, network traffic, sensor behavior, and protocol-level details, making them highly informative for multi-modal learning and ensemble methods. The dataset is distributed in CSV and PCAP formats and supports both centralized and federated learning scenarios. Its volume, diversity, and labeling depth make it particularly suitable for developing scalable DL architectures such as transformer-GAN-AE. Moreover, its wide coverage of attack vectors enhances generalization and provides a robust testbed for evaluating the resilience of intrusion detection systems under complex, evolving threat conditions.

The TON_IoT dataset was developed by the Cyber Range Lab at the University of New South Wales (UNSW) and represents a unified, cross-domain collection of telemetry, system, and network-level data tailored for AI-driven cybersecurity solutions in IoT and IIoT systems. Its design combines sensor logs from physical IoT devices, operating system-level events from Windows and Linux platforms, and packet-based network traffic, offering a rich and heterogeneous source for multi-view security modeling. TON_IoT provides data in multiple formats, including CSV, TXT, and PCAP, and contains a variety of attack scenarios such as DoS, DDoS, ransomware, reconnaissance, XSS, backdoor, injection, and password attacks. The dataset includes over 21 million samples in its network stream subset alone, as well as smaller subsets for telemetry and operating system logs. The network features (42 in total) capture statistical patterns, flow attributes, and protocol behaviors. What distinguishes TON_IoT is its versatility: it supports supervised, unsupervised, and multi-task learning settings across diverse data modalities. The dataset enables fine-grained anomaly detection, cross-platform behavioral modeling, and time-aware detection strategies. Its heterogeneous composition makes it ideal for evaluating hybrid models, such as Transformer–GAN–AE, that benefit from simultaneous generative learning, feature encoding, and temporal attention across diverse input domains.

Prior to feeding the datasets into our proposed architecture, several preprocessing steps were applied to ensure consistency, quality, and compatibility across all input sources. First, redundant, incomplete, and corrupted records were removed to eliminate noise and reduce inconsistencies that could impair model performance. Following this, categorical attributes—if present—were encoded into numerical representations using label encoding or one-hot encoding, depending on the feature semantics. This step was essential for standardizing input across heterogeneous fields, particularly in datasets such as TON_IoT, which include mixed data modalities. After encoding, all numerical features were normalized using min-max scaling to ensure they fall within a uniform range [0, 1], which facilitates smoother convergence during training and prevents dominant features from overshadowing smaller-scale variables. To address class imbalance, which is prevalent across all three datasets, we applied a combination of undersampling for dominant classes and oversampling for minority attack categories using SMOTE and random sampling strategies. These steps helped stabilize learning, especially during the adversarial training phase of GANs. Finally, for sequence-based modules such as the Transformer, the datasets were segmented into fixed-length time windows to capture temporal patterns and preserve the ordering of events. Each window was transformed into an input sequence with consistent dimensions, suitable for parallel processing through the encoder. The prepared datasets were then split into training, validation, and test subsets, with stratified sampling to maintain class distributions. These preprocessing efforts ensured a clean, balanced, and model-compatible input structure for robust performance evaluation of our hybrid intrusion detection architecture.

2.2. GAN

GANs have emerged as a transformative framework in the field of generative modeling since their introduction by Goodfellow et al. in 2014, as they are designed for unsupervised learning and provide a mechanism to learn the underlying data distribution without the need for labeled samples [40]. This capability is particularly relevant in the context of intrusion detection in edge and IIoT systems, where data are often high-dimensional, heterogeneous, and heavily imbalanced. The GAN architecture is composed of two neural networks (the generator and the discriminator) that are trained simultaneously through an adversarial process. The generator G is responsible for synthesizing fake data samples from a latent noise distribution, while the discriminator D attempts to distinguish between real and synthetic inputs. The interaction between these two networks forms a two-player minimax game in which each network iteratively improves its performance in opposition to the other. Initially, the generator produces low-quality samples that the discriminator can easily classify as fake. However, as training progresses, the generator learns to model the data distribution more effectively, eventually producing outputs that are increasingly difficult for the discriminator to distinguish from real data. This dynamic leads to a powerful generative process that can synthesize realistic data samples, making GANs particularly useful for augmenting rare classes and learning complex decision boundaries [41].

The standard structure of a GAN is illustrated in Figure 1, which shows the interaction between the dataset, the generator G, and the discriminator D. The generator takes a noise vector sampled from a prior distribution and outputs synthetic data samples. These synthetic samples, alongside real data drawn from the dataset, are fed into the discriminator, which performs a binary classification to determine whether the input is real or fake. The discriminator then provides feedback to the generator in the form of gradients, which guide the generator in learning to produce increasingly realistic samples. This adversarial training process is repeated iteratively until the generator is capable of producing samples that are indistinguishable from real data [42].

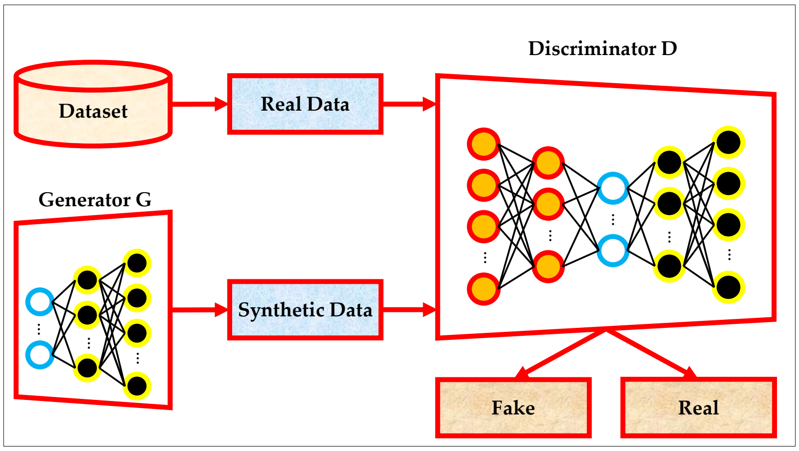


Figure 1. The architecture of standard GAN.

The objective of GAN training is captured by a value function that defines the minimax game between the generator and the discriminator. This is expressed formally in Equation (1), where the discriminator aims to maximize its ability to classify real samples as real and fake samples as fake, while the generator attempts to minimize the discriminator's ability to distinguish real from synthetic data [40]:

$$\underset{G}{min}\,\underset{D}{max}\,V(D,\,G) = \mathbb{E}_{X\sim p_{data}(x)}\,[log\,(D(X))] + \mathbb{E}_{Z\sim p_Z(z)}\,[log\,(1 - D(G(Z)))], \tag{1}$$

where $p_{data}(x)$ represents the distribution of real data; $p_Z(z)$ denotes the prior distribution over the latent noise vectors; $D(X)$ refers to the probability output of the discriminator for a real input X, while $D(G(Z))$ denotes the probability assigned to a sample synthesized by the generator. The generator and discriminator are updated alternately using stochastic gradient descent to minimize and maximize this value function, respectively. The training process for each network can be further decomposed into their individual loss functions. The loss function for the discriminator, shown in Equation (2), is designed to increase the likelihood of correctly classifying real and fake samples. It is the negative log-likelihood of assigning the correct label to both real and generated data:

$$L_D = -\Big(\mathbb{E}_{X\sim p_{data}(x)}\,[log\,(D(X))] + \mathbb{E}_{Z\sim p_Z(z)}\,[log\,(1 - D(G(Z)))]\Big), \tag{2}$$

where L_D is the loss functions for the discriminator. Conversely, the generator's objective is to deceive the discriminator by generating samples that are classified as real. The corresponding loss function, shown in Equation (3), is designed to minimize the discriminator's confidence in detecting synthetic data [41]:

$$L_G = -\mathbb{E}_{Z\sim p_Z(z)}\,[log\,(1 - D(G(Z)))], \tag{3}$$

where L_G is the loss functions for the generator. The alternating optimization of these two loss functions allows GANs to converge toward a Nash equilibrium, where the discriminator cannot distinguish real from synthetic samples with better than random chance. This mechanism allows GANs to model highly complex distributions, making them particularly effective in the context of anomaly detection in IIoT environments, where malicious behavior is rare and difficult to define using classical rules or supervised learning methods [42].

2.3. AE

AEs are unsupervised neural network architectures introduced by Rumelhart, Hinton, and Williams in the 1980s, and later popularized in DL by Hinton and Salakhutdinov in 2006. Their primary objective is to learn compact, informative representations of input data by reconstructing the input through a bottleneck architecture. The AE consists of two major components: an encoder that maps input data into a lower-dimensional latent space, and a decoder that reconstructs the input from the latent representation [40]. This compression–reconstruction mechanism enables the model to capture essential structure in the data, making AEs valuable for feature learning, dimensionality reduction, and anomaly detection. One of the key advantages of AEs is their ability to model data distributions without requiring labeled examples. This makes them highly applicable to scenarios such as intrusion detection in edge and IIoT systems, where labeled attack data are limited and often noisy. Furthermore, the reconstruction error between the input and the output can serve as a strong indicator of anomalous behavior—data points that significantly deviate from the learned distribution typically yield high reconstruction error. This capability allows AEs to operate effectively in highly imbalanced environments by learning compact representations of normal patterns and flagging deviations [43].

The architecture of a standard AE is shown in Figure 2, where real data samples (either directly from the dataset or generated by a trained GAN) are fed into an encoder–decoder pair. The encoder compresses the input into a latent feature space, and the decoder attempts to reconstruct the original input from this compressed representation. The training process involves minimizing the reconstruction loss between the input and output data. As illustrated, the encoder and decoder are composed of fully connected layers, and the overall objective is to learn a smooth, low-dimensional manifold of the input space that preserves the critical structural information [44].

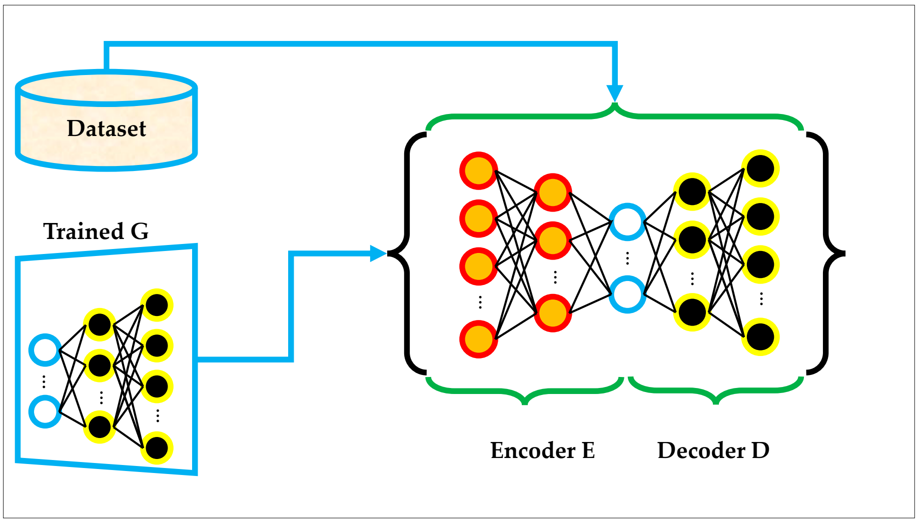


Figure 2. The architecture of standard AE.

The fundamental objective of training an *AE* is to minimize the difference between the input data and its reconstruction. This can be expressed by the reconstruction loss function shown in Equation (4):

$$L_{AE} = \frac{1}{N}\sum_{i=1}^{N}\|x_i - \hat{x}_i\|^2 \tag{4}$$

where N is the total number of input samples; x_i is the i-th input data sample; $\hat{x}_i$ is the reconstruction of x_i produced by the AE; L_{AE} is the mean squared reconstruction loss over all samples; and $\|.\|^2$ is the squared Euclidean norm. This reconstruction loss is minimized during training via backpropagation, allowing the encoder to learn efficient representations of the input data while the decoder learns to invert this transformation. In the context of our work, incorporating the AE after the GAN-generated or real input serves a dual purpose: it reduces the impact of noise and improves robustness by regularizing the input space before classification or further analysis. This denoising and reconstruction capacity of AEs makes them a powerful module within our proposed Transformer–GAN–AE hybrid framework [43].

2.4. Transformer Encoder

Transformers were introduced by Vaswani et al. in 2017 [45] as an attention-based architecture to process sequential data. Unlike recurrent models such as long short-term memory (LSTM), transformers enable parallel computation and capture long-range dependencies using self-attention mechanisms [45]. The model has become foundational in natural language processing, time-series modeling, and multivariate data analysis due to its scalability, effectiveness in learning contextual relationships, and fast training convergence. The transformer encoder operates by applying self-attention to model the relationship between all positions in an input sequence. This is particularly beneficial in domains such as intrusion detection, where the underlying patterns in system logs or network packets are often dependent on global temporal context. In edge and IIoT systems, sequences of sensor readings, system calls, or traffic data are typically complex, high-dimensional, and asynchronous. The encoder's capability to capture cross-temporal and cross-feature interactions makes it ideal for learning structured representations useful for detecting anomalies or cyber threats [46–48]. The internal structure of the encoder is shown in Figure 3.

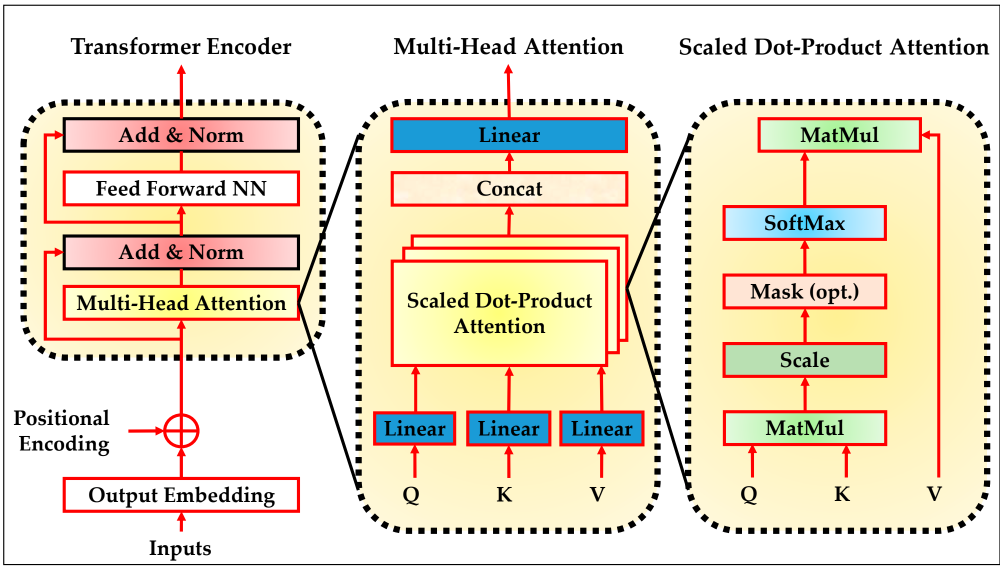


Figure 3. The transformer encoder architecture.

The process begins by embedding the input sequence and injecting positional encodings to preserve order information. Each encoder block comprises two main components: a multi-head self-attention layer and a position-wise feed-forward neural network (FFNN). These sub-layers are followed by residual connections and normalization. The self-attention mechanism enables the model to attend to different positions in the sequence simultaneously, while the feed-forward layer refines the intermediate representations. The stacked encoder layers allow for hierarchical modeling of complex dependencies over the sequence. Figure 3 also expands the core building blocks within the encoder. The scaled dot-product attention mechanism computes similarity scores between all pairs of input positions. These scores are then normalized using a softmax function to form attention weights, which are used to compute weighted combinations of the input values. This operation is applied across multiple heads in the multi-head attention module. Each head focuses on a different subspace of the input representation, and their outputs are concatenated and linearly transformed. This flexible and parallel structure enhances the model's ability to capture diverse patterns and long-range correlations, which is critical in anomaly-rich environments such as IIoT and edge deployments [47–49].

The transformer encoder begins by converting input tokens into dense vector embeddings, followed by the addition of positional encodings to retain sequential information. Since the architecture does not inherently model sequence order, sinusoidal positional encodings are applied to inject relative and absolute position information. These encodings are calculated using sine and cosine functions of different frequencies, as shown in Equations (5) and (6) [46]:

$$PE_{(pos,2i)} = sin\left(\frac{pos}{1000^{2i/d}}\right), \tag{5}$$

$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{1000^{2i/d}}\right) \tag{6}$$

Here, *pos* is the position index, *i* is the dimension index, and *d* is the embedding size. After positional encoding, the embedded sequence is passed through a series of attention layers. In each layer, the input vector is linearly projected into three matrices (queries, keys, and values), which are used in the attention calculation. These projections are computed as shown in Equations (7)–(9) [47]:

$$Q = ZW^Q, \tag{7}$$

$$K = ZW^K, \tag{8}$$

$$V = ZW^V \tag{9}$$

where W^Q, W^K, and W^V are learned projection weights, and Z is the input from the previous encoder layer (or the embedding if it is the first layer). Self-attention is then computed by measuring the similarity between queries and keys, scaled by the dimensionality of the key vectors. The resulting weights are applied to the value matrix to produce a weighted representation of the input. This process is defined in Equation (10):

$$Attention\ (Q,\ K,\ V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \tag{10}$$

Here, d_K is the dimensionality of each attention head. To enable the model to focus on different subspaces of the input, the encoder uses multi-head attention. Multiple attention outputs are concatenated and linearly projected, as described in Equation (11), and the computation for each attention head is shown in Equation (12) [49]:

$$MultiHead\ (Q,\ K,\ V) = Concat(head_1,\ head_2,\ \ldots,\ head_h)W^O, \tag{11}$$

$$head_1 = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{12}$$

W is the final output projection. The output of the multi-head attention layer is passed through a position-wise *FFNN*, which is applied independently to each position. The *FFNN* consists of two linear transformations with a ReLU activation in between, as shown in Equation (13):

$$FFNN\ (x) = \mathrm{ReLU}(0,\ xW_1, +b_1)W_2 + b_2 \tag{13}$$

where x is the input vector corresponding to a single token or position in the sequence, W is the weight matrix, and b is bias vector. Each sublayer in the encoder is followed by a residual connection and layer normalization. The output of the multi-head attention sublayer is normalized, as shown in Equation (14), and the same is done for the feed-forward output in Equation (15) [48]:

$$\acute{Z} = LayerNorm(Z + MultiHead\ (Q,\ K,\ V)), \tag{14}$$

$$Z^{out} = LayerNorm\big(\acute{Z} + FFNN\ \big(\acute{Z}\big)\big) \tag{15}$$

2.5. IChOA

The ChOA was introduced by Khishe and Mosavi in 2020 as a nature-inspired meta-heuristic optimization method based on the intelligent hunting strategies of chimpanzees [50]. The algorithm simulates the dynamic and cooperative behavior of chimps during hunting, where group members exhibit different roles including attackers, barriers, chasers, and drivers. These agents work together to balance exploration and exploitation of the search space, ensuring both global search capability and local convergence. ChOA has demonstrated strong performance in solving nonlinear, multimodal, and high-dimensional optimization problems. One of ChOA's key advantages lies in its dynamic adaptive strategy, which allows agents to transition between exploration and exploitation phases based on chaotic variables and stochastic parameters. In the exploration phase, agents are dispersed to discover global optima, while in the exploitation phase they converge towards the most promising solutions found so far. This flexible switching mechanism enables ChOA to avoid premature convergence and local optima traps, which are common issues in traditional optimization algorithms. The optimization begins by calculating the distance between a chimp agent and the prey, as described in Equation (16). Using this distance, the new position of the chimp is updated, as shown in Equation (17). These steps are governed by dynamic coefficients that control exploration and convergence pressure, defined in Equations (18)–(20) [50,51]:

$$d = \left|c.X_{prey}(t) - m.X_{chimp}(t)\right|, \tag{16}$$

$$X_{chimp}\ (t+1) = X_{\boldsymbol{prey}}\ (t) - a.d, \tag{17}$$

$$a = 2.f.\, r_1 - f, \tag{18}$$

$$c = 2.\, r_2, \tag{19}$$

$$m = Chaotic\_value \tag{20}$$

where $X_{prey}(t)$ is the prey's position vector; $X_{chimp}(t)$ denotes the chimp's position vector; r_1 and r_2 are the random vectors $\in [0,1]$; $a, c,$ and m are the coefficient vectors; m indicates a chaotic vector; and f is the dynamic vector $\in [0, 2.5]$. In each iteration, four elite chimpanzees guide the optimization process based on their hunting roles. The best chimp acts as the attacker, followed by the barrier, chaser, and driver. The distances between each of these elite agents and the current solution are computed, as shown in Equation (21). Then, four candidate updates are generated according to each elite chimp, as described in Equation (22). Finally, the next position of the agent is calculated by averaging these updates, as given in Equation (23) [52]:

$$\begin{cases} d_{Attacker} = |c_1.X_{Attacker} - m_1.X| \\ d_{Barrier} = |c_2.X_{Barrier} - m_2.X| \\ d_{Chaser} = |c_3.X_{Chaser} - m_3.X| \\ d_{Driver} = |c_4.X_{Driver} - m_4.X| \end{cases}, \tag{21}$$

$$\begin{cases} X_1 = X_{Attacker} - a_1(d_{Attacker}) \\ X_2 = X_{Barrier} - a_2(d_{Barrier}) \\ X_3 = X_{Chaser} - a_3(d_{Chaser}) \\ X_4 = X_{Driver} - a_4(d_{Driver}) \end{cases}, \tag{22}$$

$$X\ (t+1) = \frac{X_1 + X_2 + X_3 + X_4}{4} \tag{23}$$

where $X_{Attacker}$ presents the best search agent, $X_{Barrier}$ is the second-best search agent, X_{Chaser} denotes the third-best search agent, X_{Driver} is the fourth-best search agent, and $X(t+1)$ is the updated position of each chimp. The mechanism described above is visually represented in Figure 4, which illustrates how the four types of elite chimps coordinate their movements to guide the target solution. Each chimp is positioned at a certain location in the search space and exerts influence over the candidate solution using different distance vectors. The central point (R) represents the current solution, which is iteratively updated based on the directional pull of the four elite chimps. This collective behavior drives the solution towards regions of higher fitness [50].

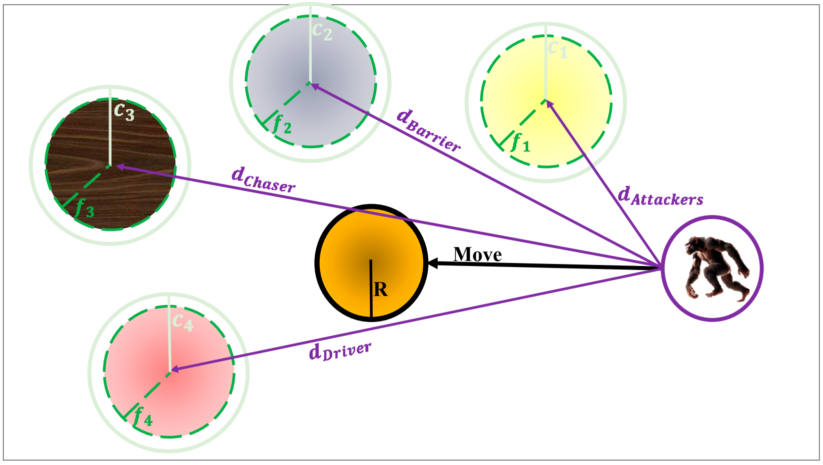


Figure 4. Position update in the ChOA algorithm.

Additionally, an adaptive variant of the distance function used in the core update process is shown in Equation (24). This modified formulation allows the algorithm to incorporate different behaviors across phases and enhance diversity when needed [50]:

$$d = \left| c.X_{prey}(t) - m.X_{chimp}(t) \right| \tag{24}$$

where μ is the random number ∈ [0, 1]. Although the standard ChOA is effective in many benchmark and real-world optimization scenarios, its performance can degrade in problems that require delicate coordination between global exploration and local exploitation. Like many population-based meta-heuristics, ChOA begins with strong exploratory behavior but gradually collapses into local search. This imbalance can lead to premature convergence, especially in complex or high-dimensional search spaces where the global optimum is surrounded by deceptive local minima. A key reason for this behavior is the lack of adaptive mechanisms that can dynamically shift the focus of the swarm based on the evolving state of the search. In its original formulation, ChOA uses four elite chimpanzee roles: Attacker, Barrier, Chaser, and Driver. The Attacker leads the swarm towards the best solution found so far, serving as the main exploitation force. The Barrier attempts to restrict movement based on observed boundaries, introducing diversity through defensive maneuvers. The Chaser tracks secondary promising candidates, while the Driver influences the swarm's general trajectory. Each role contributes a different behavioral pressure to the optimization process. However, all four agents follow fixed, deterministic roles without memory or stochastic modulation, limiting their adaptability when the search stagnates or over-converges [51,52].

To enhance the algorithm's dynamic responsiveness, we introduce a fifth elite agent: the balancer. Unlike the other roles that are biased either towards intensification or diversification, the Balancer is a hybrid agent designed to continuously monitor the swarm's state and apply a corrective influence. Its primary purpose is to maintain a balance between exploration and exploitation by dynamically adjusting its behavior. When the algorithm detects signs of stagnation or lack of diversity, the Balancer injects controlled randomness into its movement. Conversely, when convergence is meaningful and fitness improvement is steady, the Balancer shifts its trajectory closer to elite agents to intensify the local search. The Balancer accomplishes this dual function through a two-pronged mechanism. First, a portion of its decision vector is generated using chaotic or uniformly random components, which ensures that unexplored regions of the search space are periodically sampled. This stochasticity allows the Balancer to break symmetry and encourage diversity without depending on the global population structure. Second, the Balancer maintains a lightweight memory buffer of its historically best position. At each iteration, it evaluates its current fitness against its personal best and incorporates this historical knowledge as a guiding force. This memory-based guidance ensures that once a promising region is encountered, it can be revisited and refined with increased precision.

The result is a self-adaptive behavior that operates independently of the swarm's convergence level. During early stages of the search, the Balancer primarily contributes to exploration by introducing perturbations that guide the population away from early local optima. As the algorithm progresses, it gradually shifts towards exploitation by leveraging its memory and aligning with high-performing agents. Its behavior is not fixed or predefined, but instead evolves based on the swarm's convergence rate, diversity level, and fitness improvement. This allows the Balancer to act as a dynamic regulator of search pressure, filling the gap between aggressive convergence and random wandering. To integrate the proposed balancer agent into the ChOA framework, the standard position update equations have been modified accordingly. The updated distance computation now includes a fifth term, corresponding to the Balancer's influence, as shown in Equation (25). Similarly, the position updates from each elite agent are extended to include the contribution from the Balancer in Equation (26). Finally, the new position of the agent is determined by averaging the effects of all five elite chimps, resulting in the revised population update rule presented in Equation (27). This extension allows the algorithm to dynamically adapt its search behavior by blending directional guidance from both historical best agents and adaptive exploration driven by the Balancer's hybrid mechanism.

$$\begin{cases} d_{Attacker} = |c_1.X_{Attacker} - m_1.X| \\ d_{Barrier} = |c_2.X_{Barrier} - m_2.X| \\ d_{Chaser} = |c_3.X_{Chaser} - m_3.X| \\ d_{Driver} = |c_4.X_{Driver} - m_4.X| \\ d_{Balancer} = |c_5.X_{Balancer} - m_5.X| \end{cases}, \quad (25)$$

$$\begin{cases} X_1 = X_{Attacker} - a_1(d_{Attacker}) \\ X_2 = X_{Barrier} - a_2(d_{Barrier}) \\ X_3 = X_{Chaser} - a_3(d_{Chaser}) \\ X_4 = X_{Driver} - a_4(d_{Driver}) \\ X_5 = X_{Balancer} - a_5(d_{Balancer}) \end{cases}, \tag{26}$$

$$X\ (t+1) = \frac{X_1 + X_2 + X_3 + X_4 + X_5}{5} \tag{27}$$

Integrating the balancer into the ChOA framework introduces a dynamic self-regulation mechanism that enhances the optimizer’s adaptability to complex, high-dimensional search landscapes. This capability is particularly crucial for tuning the hyperparameters of deep hybrid models such as the proposed Transformer–GAN–AE architecture, where the performance is highly sensitive to initialization choices, learning rates, latent dimensions, and architectural configurations. The Balancer enables the algorithm to maintain diversity during early stages of the search while reinforcing exploitation near promising solutions, without requiring predefined switching thresholds. As a result, the improved ChOA can more effectively navigate the intricate and multimodal hyperparameter space, offering more stable convergence behavior, superior model generalization, and higher-quality configurations for robust intrusion detection in Edge and IIoT environments.

2.6. Optimized Transformer-GAN-AE

The proposed Transformer–GAN–AE architecture, illustrated in Figure 5, presents a hybrid DL framework designed to address the challenges of anomaly detection in Edge and IIoT environments. It integrates three powerful modules each selected for its complementary strengths in handling high-dimensional, imbalanced, and noisy intrusion data. These modules are interconnected through a coordinated training pipeline, where outputs from one component refine or enhance the inputs to the next. The entire architecture is further optimized by an IChOA to automatically tune critical hyperparameters, improving generalization, convergence, and robustness. At the front end of the system is the transformer encoder, responsible for modeling sequential dependencies and extracting contextualized temporal features from raw network traffic or sensor logs. The encoder processes embedded inputs enriched with positional encoding and captures long-range relationships using multi-head self-attention mechanisms. This allows the model to learn intricate patterns that span across time and protocol layers, making it highly effective in modeling evolving behavior patterns within IIoT traffic streams. The Transformer is particularly valuable for detecting subtle, slow-evolving attacks that span across time windows.

Next, a GAN module is introduced to enhance representation learning and improve class balance. The generator is trained to synthesize realistic data samples that resemble normal behavior patterns, while the discriminator learns to distinguish between real and synthetic samples. This adversarial learning paradigm forces the generator to produce samples that align closely with the real distribution, effectively augmenting minority classes and refining boundary regions. GAN-generated data are later used to regularize the training of the AE and Transformer components by improving their generalization under data imbalance. Following the GAN, the data (either original or GAN-augmented) is passed to the AE module, which functions as a denoising and anomaly-sensitive encoder-decoder network. The AE compresses the input through its encoder and reconstructs it via the decoder.

The reconstruction error serves as an indicator of anomalous behavior. Because the AE is trained primarily on normal data, its ability to accurately reconstruct malicious or unusual inputs is reduced, making it a reliable signal for anomaly detection. The AE thus plays a vital role in feature refinement and noise suppression prior to the final decision layers. The interplay between the three modules is a key design feature of the proposed architecture. The Transformer ensures contextual learning of temporal dynamics, the GAN balances data distributions and enriches representational capacity, and the AE acts as a final filter that highlights deviations from learned norms. This combination enables the system to robustly detect both known and unknown intrusions across varying attack surfaces. The diversity in their operational principles (attention-based, generative, and reconstruction-focused) results in a more comprehensive and fault-tolerant detection framework.

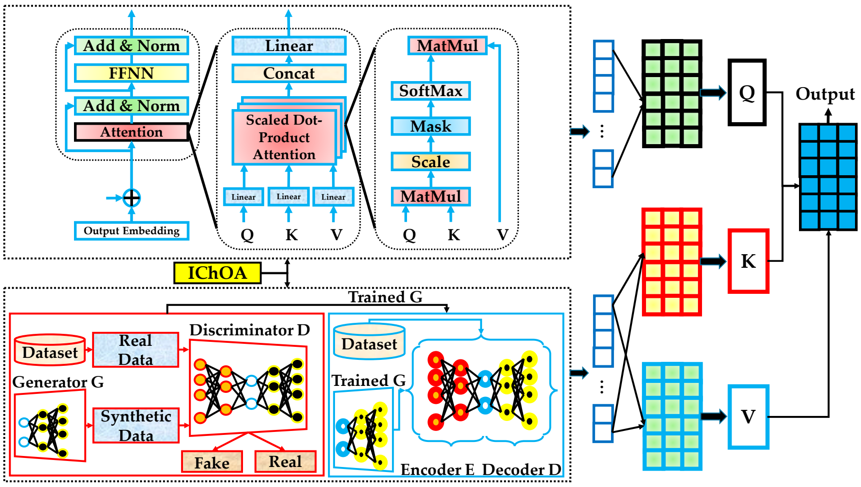


Figure 5. The proposed Transformer–GAN–AE architecture.

The design follows a sequential yet functionally complementary structure, where each component addresses a specific challenge in intrusion detection: temporal modeling, class imbalance, and feature denoising, respectively. First, the input sequence X, representing raw traffic data or sensor measurements with sequence length L and feature dimension d, is passed to the Transformer encoder. The Transformer models temporal dependencies and contextual relationships, producing a high-dimensional embedding representation as shown in Equation (28):

$$H = Transformer(X) \tag{28}$$

where $X \in \mathbb{R}^{L\times d}$ is the original sequence; $H \in \mathbb{R}^{L\times d'}$ is the temporal embedding output; L is the sequence length; d is the input feature dimension; d' is the embedding size after transformation; $Transformer(.)$ denotes the full encoder stack applied to the sequence.

Second, the output H is used to train a GAN. The generator synthesizes realistic feature representations to address class imbalance, while the discriminator evaluates their plausibility. The output of this process is a set of synthetic samples X_{GAN}, defined in Equation (29):

$$X_{GAN} = GAN(H) \tag{29}$$

where $GAN(.)$ denotes the generative process conditioned on real embeddings; X_{GAN} is the generated feature sequence mimicking the real embedding structure H. Third, the embeddings from both real and generated sources are concatenated and passed through an AE to compress and denoise the joint representation. This step is captured in Equation (30):

$$Z = AE([H, X_{GAN}\ ]) \tag{30}$$

where $[H, X_{GAN}\ ]$ denotes the concatenation of real and generated embeddings along the batch dimension; $AE(.)$ is the AE (encoder + decoder); Z is the compressed, denoised latent representation used for downstream classification.

A critical contribution of the proposed architecture lies in its optimization backbone, driven by an enhanced IChOA. IChOA is used to automatically tune the hyperparameters of all three modules, addressing a major bottleneck in deep learning deployment: manual configuration. Hyperparameters such as the number of attention heads, embedding dimension, and feedforward width in the transformer, the latent dimension and learning rate in the GAN, and the layer width in the AE are all optimized via IChOA. Additionally, IChOA optimizes the initial weights and biases to accelerate convergence and avoid poor local minima. The reason hyperparameter tuning is critical in this setup is due to the high sensitivity of deep models to architectural settings. Improper configuration can lead to vanishing gradients, unstable adversarial training, overfitting, or inefficient learning. The stochastic nature of GAN training, in particular, makes it highly dependent on well-calibrated learning rates, loss balancing coefficients, and latent noise shapes. The transformer's performance hinges on embedding dimensionality and depth, while AE's sensitivity to reconstruction errors depends on how tightly its bottleneck is tuned. Manual tuning of these settings is time-consuming and often suboptimal.

IChOA enhances ChOA by introducing a fifth chimp role which adaptively regulates exploration and exploitation during optimization. This ensures that the hyperparameter search space is explored sufficiently in the early stages and exploited effectively in the later stages. The balancer uses a memory-guided and partially random strategy to introduce diversity without destabilizing convergence. This mechanism is particularly useful in hyperparameter tuning, where local traps and deceptive fitness landscapes are common. By distributing search pressure more effectively, IChOA improves the quality of selected configurations while maintaining algorithmic stability. By integrating IChOA at the top level of the architecture, the system becomes self-optimizing. During training, IChOA evaluates candidate hyperparameter sets based on validation accuracy or reconstruction error, and iteratively updates them to improve performance. This optimization loop wraps around the entire transformer–GAN–AE pipeline, guiding each module toward its optimal operational state. The result is an adaptive and automated system capable of generalizing across datasets and evolving attack vectors with minimal manual intervention.

3. Results

All experiments were conducted using a workstation equipped with an Intel Core i9-12900K CPU, 64 GB of RAM, and an NVIDIA RTX 3090 GPU with 24 GB of VRAM, running on Ubuntu 22.04 LTS. The implementation of the proposed transformer–GAN–AE framework and all baseline models was carried out in Python 3.9, utilizing PyTorch 2.0.1 as the primary DL library. Data preprocessing, scaling, and augmentation were performed using Pandas 1.5.3 and Scikit-learn 1.2.2, while numerical operations relied on NumPy 1.24.2. Visualization of results was handled using Matplotlib 3.7.1 and Seaborn 0.12.2. Experiments were conducted independently on three benchmark datasets—WUSTL-IIoT-2021, Edge-IIoTset, and TON_IoT, using a consistent train-validation-test split of 70%, 15%, and 15%, respectively. Each experiment was repeated 25 times, and the average results are reported to ensure statistical robustness.

To validate the effectiveness of the proposed Transformer–GAN–AE framework, we conducted comprehensive comparisons against a set of well-established baseline models spanning generative learning, deep neural architectures, and gradient-based classifiers. These include the standard GAN, convolutional neural network (CNN), deep belief network (DBN), time-series transformer (TST), bidirectional encoder representations from transformers (BERT), and extreme gradient boosting (XGBoost). This diverse selection ensures a thorough and fair evaluation across multiple architectural paradigms, covering both DL and ML perspectives. The inclusion of GAN as a baseline enables a direct comparison with the generative component of our framework, isolating the value added by the integrated AE and Transformer modules. CNN is widely adopted in intrusion detection tasks due to its efficiency in capturing spatial hierarchies and local patterns within feature vectors, making it a strong convolutional benchmark. DBN, as a classical unsupervised deep architecture, allows us to evaluate performance from a probabilistic representation learning perspective, particularly under noisy or sparse data conditions. Its hierarchical structure and pre-training mechanism provide useful insight into how deep generative models handle IIoT data.

For temporal modeling, we include TST and BERT, both of which leverage the self-attention mechanism but differ in complexity and design philosophy. TST represents a lightweight temporal transformer tailored for time series forecasting and sequence classification, making it highly relevant for capturing IIoT traffic patterns. BERT, on the other hand, brings contextualized embeddings and bidirectional attention modeling to the task, offering a powerful comparison point for language-model-inspired architectures applied to security data. Finally, XGBoost is selected as a non-deep learning benchmark due to its exceptional performance in many tabular intrusion detection problems. As a powerful ensemble method with gradient boosting, XGBoost serves as a strong classical baseline for evaluating the added value of deep and hybrid models. The inclusion of these models provides a holistic benchmark space to assess generalization, anomaly sensitivity, and robustness across architectures.

To assess the effectiveness of the proposed Transformer–GAN–AE architecture and baseline models in detecting intrusions, we employed several widely recognized performance metrics. First, accuracy was used to measure the overall correctness of the model's predictions. As shown in Equation (31), accuracy is defined as the ratio of correctly classified instances (*true positives* and *true negatives*) to the total number of predictions. While accuracy provides a general sense of performance, it can be misleading in highly imbalanced datasets, where the dominant class skews the metric. However, in this study, it was still a useful indicator when combined with more sensitive measures.

$$Accuracy = \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negative} \tag{31}$$

To address class imbalance and provide a more nuanced view of detection capability, recall (also known as sensitivity or *true positive* rate) was utilized. Equation (32) defines recall as the proportion of actual positive cases correctly identified by the model. This metric is particularly important in intrusion detection scenarios, where false negatives (i.e., undetected attacks) can result in severe security breaches. High recall ensures that the model effectively identifies attack instances, which is critical in minimizing risk in IIoT and edge deployments.

$$Recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{32}$$

We further evaluated the models using the area under the curve (*AUC*), as described in Equation (33). *AUC* provides a comprehensive view of the model's discriminatory ability across all possible threshold settings. By integrating the receiver operating characteristic (*ROC*) curve over all classification thresholds, this metric captures the trade-off between *true positive* and *false positive* rates. *AUC* is particularly valuable in binary and multiclass intrusion detection, as it is less affected by class distribution and reflects the model's robustness across different operating conditions.

$$AUC = \int_0^1 ROC(t)dt \tag{33}$$

where $ROC(t)$ is ROC curve at threshold t. To analyze the training behavior and convergence characteristics of each model, we also computed the root mean square error (*RMSE*) during training epochs, as given in Equation (34). *RMSE* quantifies the deviation between predicted values and actual values in terms of their squared differences. In the context of our model, *RMSE* was applied to monitor the reconstruction quality of the *AE* and the output of the generator network. Lower *RMSE* indicates better convergence and learning stability, providing insight into whether the model is effectively minimizing reconstruction and adversarial loss over time.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{y}_i]^2}, \tag{34}$$

where y_i is the observed value and $\hat{y}_i$ is the calculated value. We also examined the variance of results across multiple independent runs and the convergence behavior during training to evaluate the robustness and stability of each model. Variance reflects the degree of sensitivity a model has to changes in initialization, stochastic components (e.g., random sampling, noise vectors), or hyperparameter tuning. A lower variance across repeated executions indicates that the model's performance is stable and less prone to randomness, which is particularly important in security-critical applications where predictability and reproducibility are essential. Furthermore, the convergence trend, as monitored through metrics such as *RMSE* and loss curves over epochs, provides insight into the training dynamics. Smooth and rapid convergence suggests that the model is learning effectively and optimizing the objective functions efficiently. In contrast, oscillatory or plateaued convergence patterns can reveal issues such as vanishing gradients, adversarial instability, or suboptimal parameter initialization. In addition to the above metrics, we recorded the execution time of each model as a proxy for computational efficiency and measured the variance of performance across multiple experimental runs to evaluate consistency and generalizability. These supplementary criteria help assess not only the accuracy of predictions but also the practical feasibility and stability of each method when deployed in resource-constrained or real-time environments.

Hyperparameter tuning plays a critical role in determining the final performance and stability of Dl models, especially in complex hybrid architectures such as Transformer–GAN–AE. Without proper tuning, deep models can suffer from underfitting, overfitting, or unstable convergence—issues that directly impact classification accuracy and generalization. Parameters such as learning rate, dropout rate, hidden layer size, activation functions, and transformer depth significantly influence the training dynamics and decision boundaries. In the proposed model, the optimization of these hyperparameters was handled using the IChOA, which adaptively explored the parameter space through a dynamic balance of exploitation and exploration. This allowed the model to reach near-optimal configurations that enhanced convergence speed, detection performance, and architectural stability across diverse IIoT datasets. Table 1 summarizes the tuned hyperparameters for the proposed Transformer–GAN–AE and all baseline models. The proposed model employed IChOA for fine-tuning key architectural components such as the number of attention heads (8), encoder layers (6), feedforward hidden size (2048), and dropout rate (0.2).

The IChOA optimizer dynamically regulated learning rate (0.003), convergence threshold (0.059), momentum (0.03), and latent dimensions, resulting in a balanced and stable configuration. In contrast, the baseline models, including GAN, CNN, DBN, TST, BERT, and XGboost, were tuned using grid search, a brute-force method that systematically tests combinations from a predefined set of hyperparameter values. While grid search is simple and widely used, it is computationally expensive and lacks adaptive search behavior, often missing fine-grained optima that stochastic optimizers such as IChOA can capture.

Table 1. Parameter setting of proposed methods.

Model	Parameter	Value
Transformer-GAN-AE	Learning rate	0.003
	Batch size	32
	Feed forward hidden size	2048
	Weight decay	0.01
	Dropout rate	0.2
	Number of attention heads	8
	Number of encoder layers	6
	Activation function	GELU
	Optimizer	IChOA
	Momentum term	0.03
	Convergence threshold	0.059
	Hidden Layer Sizes	60
	a	$[-1, 1]$
	Population size	100
	Iteration	300
GAN	Learning rate	0.005
	Number of neurons in hidden layers	32
	Batch size	64
	Momentum term	0.06
	Activation function	ReLU
	Convergence threshold	0.063
	Optimizer	SGD
BERT	Learning rate	0.002
	Batch size	32
	Dropout rate	0.1
	Number of self-attention heads per layer	6
	Number of transformer encoder layers	6
	Length of input time-series window	64
	Activation Function	GELU
	Optimizer	SGD
TST	Learning rate	0.037
	Number of attention heads	6
	Number of transformer layers	6
	Dropout rate	0.2
	Optimizer	Adam
CNN	Number of convolution layers	10
	Kernel size	5 × 5
	Pooling type	Max pooling (2 × 2)
	Number of neurons	64
DBN	Number of hidden layers	6
	Number of neurons in hidden layers	32
	Learning rate	0.009
	Activation	Tanh and sigmoid
	Optimizer	SGD
XGBoost	Learning rate	0.29
	Max depth	6
	Number of estimators	300

A closer look at Table 1 reveals distinct patterns. For instance, the GAN baseline achieved its best performance using a learning rate of 0.005, a batch size of 64, and ReLU activation with SGD optimization. TST benefited from a relatively high learning rate (0.037) and Adam optimizer, reflecting its sensitivity to gradient flow in attention mechanisms. BERT was tuned with conservative parameters, including a lower dropout rate (0.1) and

SGD, which is consistent with its stable, deep encoding nature. CNN's architecture was fixed at 10 convolutional layers with 5 × 5 kernels and 64 neurons, leveraging max pooling for local spatial abstraction. DBN used six hidden layers with 32 neurons each and tanh-sigmoid activation under SGD, while XGboost achieved optimal results using a learning rate of 0.29, depth of 6, and 300 estimators, highlighting its fast convergence and ensemble strength. These tuned configurations allowed for a fair and competitive benchmarking environment against the optimized Transformer–GAN–AE model.

In Table 2, the performance of the proposed Transformer–GAN–AE model is compared against six well-established baseline algorithms, including BERT, TST, DBN, GAN, CNN, and XGBoost. The table presents three key classification metrics (accuracy, recall, and AUC) evaluated across all three datasets. This comparative analysis captures not only the general correctness of each model (accuracy) but also its ability to detect attack instances (recall) and its overall discriminative power across decision thresholds (AUC). Together, these metrics offer a robust and comprehensive understanding of each model's detection performance and generalization ability in IIoT security settings. The proposed Transformer–GAN–AE consistently outperforms all baseline models across all metrics and datasets. On the WUSTL-IIoT-2021 dataset, it achieves an accuracy of 97.86%, a recall of 98.63%, and an AUC of 99.18%, surpassing the next best model (BERT) by more than 6.5% in accuracy and nearly 6 points in AUC. A similar pattern is observed in Edge-IIoTset, where Transformer–GAN–AE achieves the highest scores in all three metrics: 98.63% accuracy, 98.79% recall, and 99.53% AUC. On the most challenging and diverse dataset, TON_IoT, the model again outperforms the alternatives with 98.92% accuracy, 99.52% recall, and a near-perfect AUC of 99.87%. These results clearly demonstrate the superiority of the proposed model in both balanced and imbalanced settings.

Table 2. Comparative performance of the proposed model and other methods across three datasets.

Model	WUSTL-IIoT-2021			Edge-IIoTset			TON_IoT		
	Accuracy	Recall	AUC	Accuracy	Recall	AUC	Accuracy	Recall	AUC
Transformer-GAN-AE	97.86	98.63	99.18	98.63	98.79	99.53	98.92	99.52	99.87
BERT	91.36	92.08	93.47	90.24	91.20	92.54	92.38	93.61	94.51
TST	89.29	90.34	91.48	90.35	91.96	93.07	90.18	91.78	92.36
DBN	87.35	88.49	89.20	88.26	89.29	90.28	91.27	92.69	93.79
GAN	85.64	85.69	87.18	86.31	87.26	88.96	89.16	90.65	90.08
CNN	83.19	84.17	85.68	85.41	86.92	88.47	86.31	87.09	89.14
XGBoost	80.18	81.24	82.74	81.60	82.76	83.19	81.64	83.61	83.17

This performance advantage can be attributed to several architectural strengths. The transformer encoder effectively captures temporal dependencies and contextual patterns in time-series intrusion data, while the GAN component synthesizes realistic minority class samples, reducing the impact of data imbalance. The AE, placed downstream, enhances anomaly sensitivity through reconstruction-based filtering. The integration of these three components allows the model to learn both generative and discriminative representations, which leads to stronger decision boundaries and higher recall in detecting subtle or rare attack patterns. In contrast, single-module baselines such as CNN or DBN lack the composite learning capacity needed for multi-layered intrusion behaviors. Furthermore, the use of the IChOA for hyperparameter tuning has contributed significantly to the model's stability and generalization. Unlike the grid search used for baseline models, IChOA adaptively explores the search space and fine-tunes key parameters such as attention head count, latent dimension size, and learning rate. This optimization ensures that the model converges rapidly and remains robust across datasets. The results confirm that Transformer–GAN–AE is not only more accurate but also more resilient to data variability, making it an ideal solution for deployment in real-world IIoT and Edge-based intrusion detection scenarios.

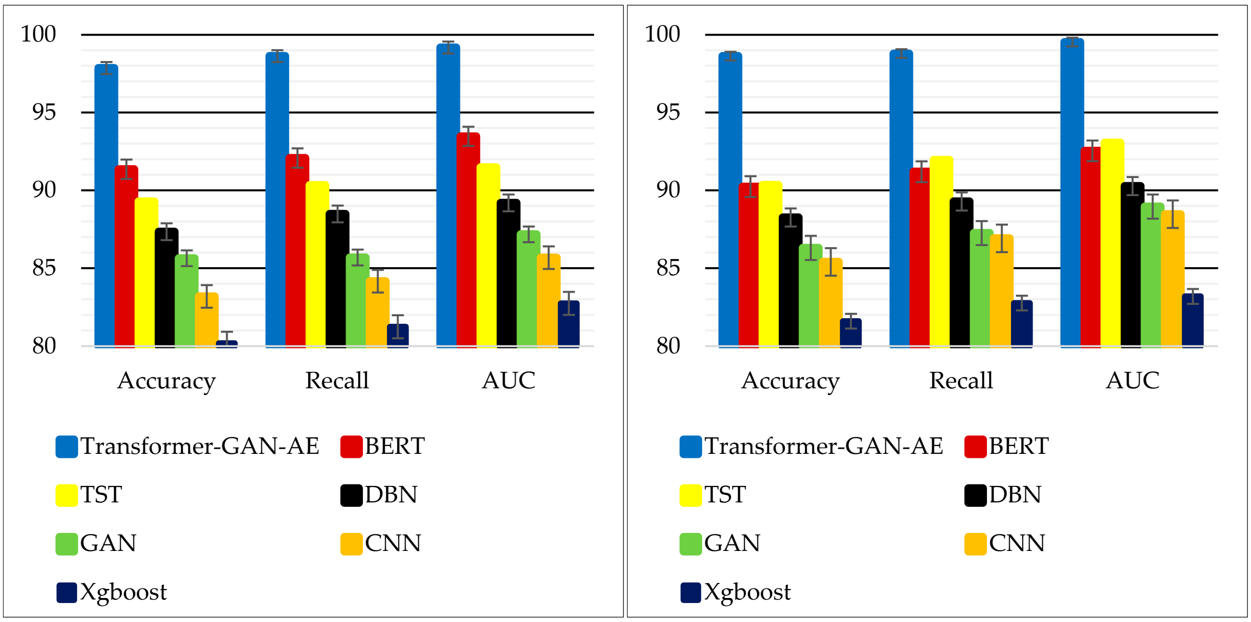
To further investigate the effectiveness of the proposed IChOA optimizer beyond our main framework, Table 3 presents the performance of multiple baseline architectures (namely BERT, TST, DBN, GAN, and CNN) when optimized using IChOA rather than traditional tuning methods. The goal is to isolate the impact of IChOA across different models and understand whether the observed improvements in our model are purely due to optimization or stem from deeper architectural synergy. The results in Table 3 demonstrate that applying IChOA led to consistent performance improvements across nearly all baseline models compared to their original versions in Table 2. For example, BERT's accuracy improved from 91.36% to 92.03% on WUSTL-IIoT-2021 and from 92.38% to 93.96% on TON_IoT. Similarly, TST showed significant gains, reaching 92.29% accuracy on TON_IoT, compared to 89.29% previously. These improvements confirm that IChOA is an effective optimizer for deep and hybrid models in complex, high-dimensional settings such as IIoT intrusion detection.

Table 3. Comparative performance of all models after IChOA-based hyperparameter optimization.

Model	WUSTL-IIoT-2021			Edge-IIoTset			TON_IoT		
	Accuracy	Recall	AUC	Accuracy	Recall	AUC	Accuracy	Recall	AUC
Transformer-GAN-AE	97.86	98.63	99.18	98.63	98.79	99.53	98.92	99.52	99.87
Transformer-CNN-AE	92.34	93.65	93.89	93.25	93.98	94.24	93.08	93.59	94.29
Transformer-DBN-AE	92.89	94.05	94.21	93.89	94.32	94.09	94.01	94.82	95.67
BERT-IChOA	92.03	93.20	94.12	91.25	92.34	93.32	93.96	94.42	95.41
TST-IChOA	90.48	91.18	92.27	91.76	92.70	93.15	92.29	93.93	94.89
DBN-IChOA	88.46	89.40	90.33	89.43	90.12	91.20	92.76	93.87	94.40
GAN-IChOA	86.41	86.77	88.34	87.16	88.81	89.32	90.27	91.50	91.28
CNN-IChOA	84.25	85.32	86.46	86.40	87.70	89.76	87.63	88.28	90.36

Despite these enhancements, none of the IChOA-augmented baseline models surpassed the proposed Transformer–GAN–AE architecture, which still maintained the highest accuracy, recall, and AUC across all three datasets. This suggests that while optimization plays a critical role in improving model performance, the synergistic interaction between the Transformer, GAN, and AE modules in our architecture provides a unique advantage, particularly in capturing temporal dependencies, balancing class distributions, and denoising feature representations. Additionally, Table 3 includes two architectural variants (Transformer-CNN-AE and Transformer-DBN-AE) that replace the GAN component with CNN and DBN, respectively. Although both models achieved relatively high scores, they consistently lagged behind the full Transformer–GAN–AE configuration. This highlights the importance of the GAN's generative capacity in enhancing minority class representation and further confirms the necessity of all three components working in tandem for optimal detection performance.

Figure 6 presents a graphical summary of the model performance metrics shown in Table 2, highlighting differences across architecture types concerning accuracy, recall, and AUC. Subfigure (a) corresponds to results on the WUSTL-IIoT-2021 dataset, while subfigure (b) depicts the same metrics on Edge-IIoTset. Each cluster of bars compares the performance of the proposed Transformer–GAN–AE with six baseline models: BERT, TST, DBN, GAN, CNN, and XGBoost. The visual format accentuates performance margins, making it easier to observe relative advantages and stability across evaluation criteria. As shown, the Transformer–GAN–AE significantly outperforms all competing models across all metrics and datasets, maintaining values near or above 98% in both accuracy and recall, with AUC scores approaching 100%. These consistently high bars reflect the model's ability to balance precision and recall, while also achieving superior discriminatory power across thresholds. In contrast, BERT and TST form a second tier of performance, with noticeably lower AUC and recall values. The models based on CNN, DBN, and XGBoost occupy the lowest tier, particularly in AUC, which highlights their limited capacity to differentiate attack from benign behavior under varied conditions. The visual clarity offered by Figure 6 reinforces the superiority of the proposed architecture and its ability to maintain robustness across complex IIoT datasets.



(**a**) (**b**)

Figure 6. Visual comparison of detection performance across models: (**a**) WUSTL-IIoT-2021; (**b**) Edge-IIoTset.

Figure 7 illustrates a radar plot that visually compares the performance of the proposed Transformer–GAN–AE architecture against six baseline models. In radar charts, each axis represents a separate metric, and the plotted area corresponds to a model's magnitude in that metric. Larger, outward-spreading shapes indicate stronger overall performance. This visualization is particularly effective for multi-criteria comparisons as it simultaneously captures the balance and dominance of each model across multiple axes. From the chart, it is evident that Transformer–GAN–AE forms the largest and most balanced polygon, nearly saturating all three axes with values above 98%, indicating that it dominates consistently in terms of detection correctness, sensitivity, and discriminative power. In contrast, models such as XGBoost and CNN form much smaller and more asymmetric regions, suggesting weaker and less uniform performance across metrics. BERT and TST perform relatively well, but still fall short of the proposed model in terms of AUC and recall, especially on the high-dimensional, imbalanced TON_IoT dataset. The radar plot confirms the robustness of Transformer–GAN–AE under complex IIoT conditions, where maintaining high values across all evaluation metrics is critical. Models with irregular or concave polygons are more prone to trade-offs, performing well on one metric while underperforming on others. The ability of the proposed model to form a near-perfect, outward-reaching triangle suggests its superior optimization, balanced architecture, and improved generalization. This reinforces the numerical findings from Table 2 and supports the model's suitability for real-world deployment in mission-critical environments.

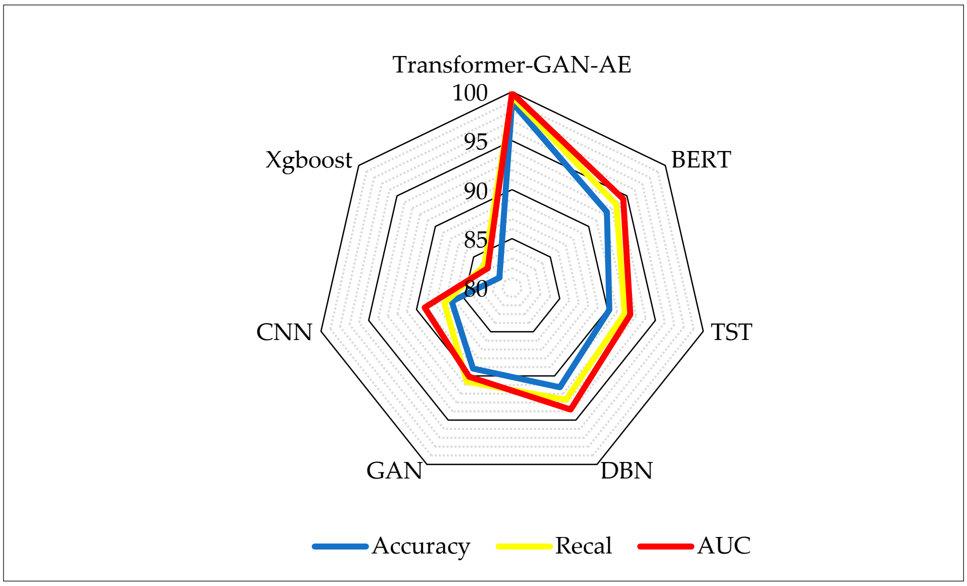


Figure 7. Radar chart comparison of model performance on the TON_IoT dataset.

Figures 8–10 present the ROC curves for all models across the three benchmark datasets. Each ROC curve plots the true positive rate (sensitivity) against the false positive rate (1-specificity), capturing the performance of a model at various decision thresholds. The closer a curve approaches the top-left corner of the plot, the higher the true positive rate and the lower the false positive rate, which indicates superior classification performance. ROC curves are particularly valuable in security applications, as they reveal a model's ability to detect attacks under varying confidence levels and class imbalance. In all three figures, the Transformer–GAN–AE model consistently dominates, forming curves that hug the top-left boundary of the plots (resulting in a high AUC), nearly reaching 1.0. On the WUSTL-IIoT-2021 dataset (Figure 8), it sharply outpaces other models with early and sustained sensitivity, while TST and BERT form the next best group, lagging behind in the midrange thresholds.

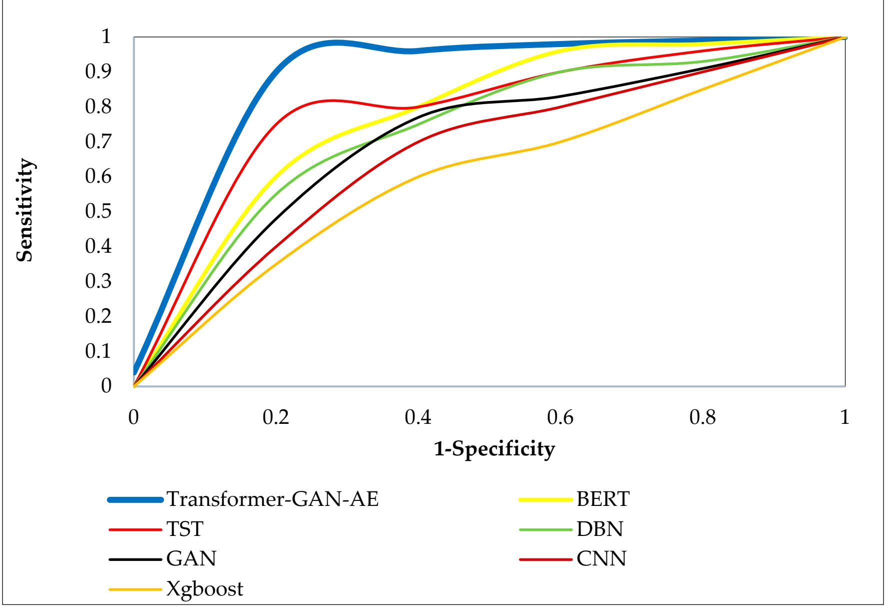


Figure 8. ROC curves of all evaluated models on the WUSTL-IIoT-2021 dataset.

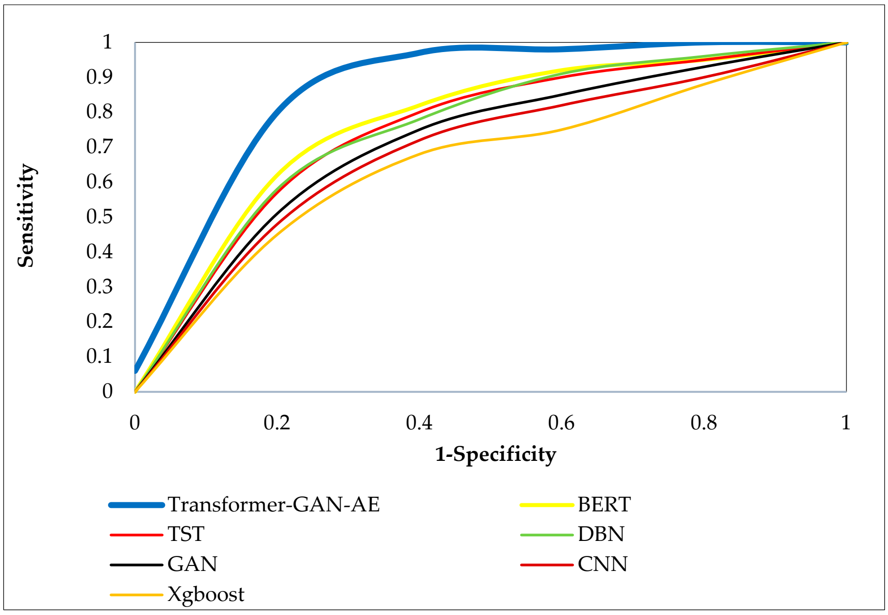


Figure 9. ROC curves of all evaluated models on the Edge-IIoTset dataset.

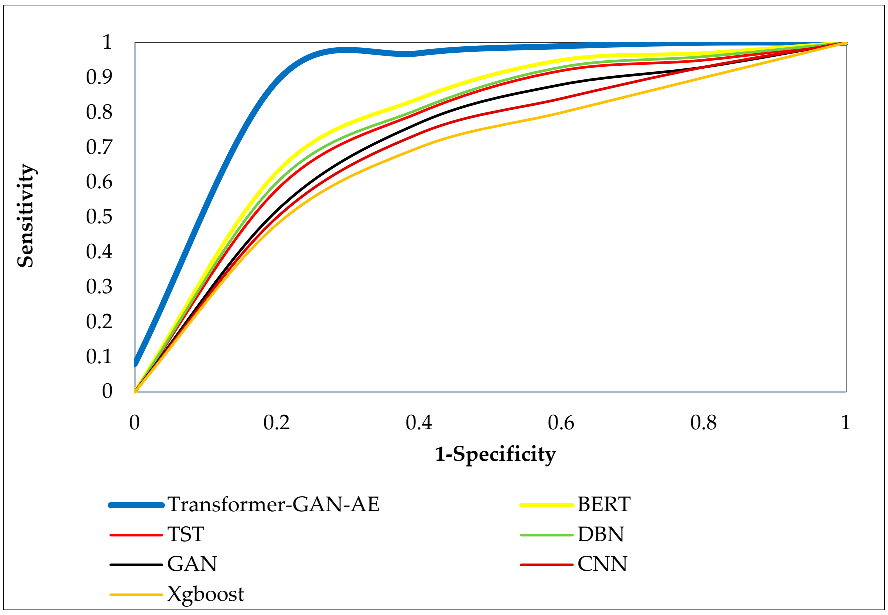


Figure 10. ROC curves of all evaluated models on the TON_IoT dataset.

On the Edge-IIoTset dataset (Figure 9), the performance trend remains consistent, though the margins between models slightly narrow. Nevertheless, Transformer–GAN–AE still maintains the most favorable ROC shape, confirming its robustness in highly dynamic and layered environments. Similar trends are evident in Figure 10 (TON_IoT), where again the proposed model demonstrates a significantly steeper and more optimal curve compared to the baselines. These visualizations reinforce the conclusion that Transformer–GAN–AE not only achieves higher classification metrics but does so with greater stability and generalization across varying detection thresholds. The baseline models show more gradual curves (particularly GAN, CNN, and XGBoost), indicating poorer sensitivity at lower false positive rates. These insights underline the architectural advantages of combining generative augmentation, temporal attention, and feature denoising, supported by IChOA-driven optimization, all of which contribute to the proposed model's consistently superior ROC behavior.

Figures 11–13 present the RMSE convergence trends for all evaluated models during training on the WUSTL-IIoT-2021, Edge-IIoTset, and TON_IoT datasets, respectively. These curves visualize the model's learning behavior and optimization efficiency across 300 epochs. RMSE quantifies the difference between predicted and actual values, so a lower value indicates more accurate predictions and a more successful training phase. The faster and smoother a model's RMSE declines, the more stable and efficient its convergence process is. In all three figures, the proposed Transformer–GAN–AE demonstrates a sharp and early RMSE drop, stabilizing near zero well before the 100th epoch and maintaining that performance throughout training.

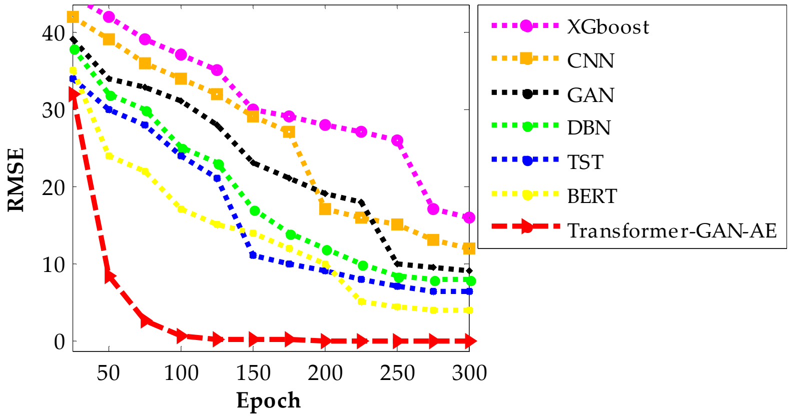


Figure 11. RMSE convergence curves of all models on the WUSTL-IIoT-2021 dataset.

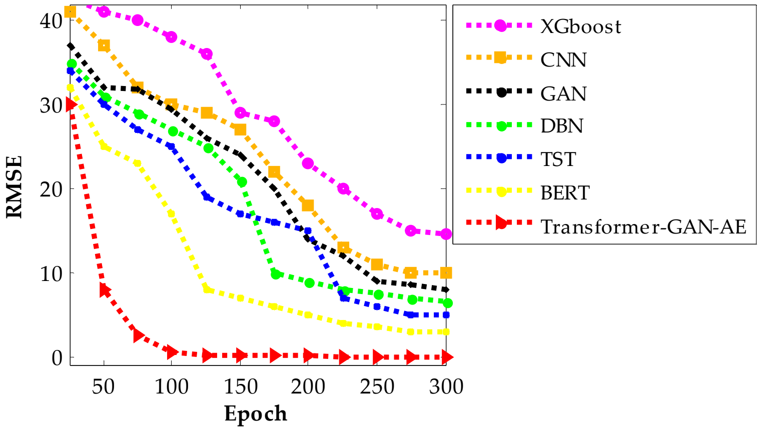


Figure 12. RMSE convergence curves of all models on the Edge-IIoTset dataset.

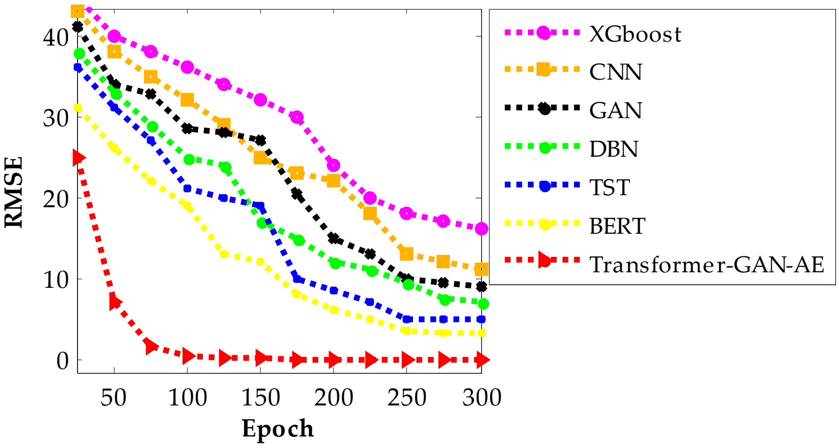


Figure 13. RMSE convergence curves of all models on the TON_IoT dataset.

This contrasts with the baselines, especially XGBoost, CNN, and GAN, which show slower, noisier, and often incomplete convergence. Even the better-performing deep models like TST and BERT exhibit occasional fluctuations or plateaus, particularly under complex datasets such as TON_IoT. The superior convergence of Transformer–GAN–AE is a direct result of both architectural synergy and the use of the IChOA optimizer. A key contributor to this behavior is the inclusion of the Balancer chimp agent in the improved IChOA algorithm. Unlike conventional optimizer strategies that often overcommit to either exploration or exploitation, the balancer injects controlled randomness and memory-based refinement into the search process. This hybrid behavior helps avoid premature convergence and stagnation in local minima. As a result, Transformer–GAN–AE benefits from a more adaptive and stable learning trajectory. The visual evidence from Figures 11–13 reinforces that the proposed model not only learns faster but also generalizes better, driven by strategic optimizer design and architectural robustness.

4. Discussion

The proposed Transformer–GAN–AE framework demonstrates substantial potential beyond numerical accuracy metrics. Its hybrid architecture not only excels in detection capability but also integrates three essential characteristics (adaptability, generative robustness, and temporal modeling) that make it particularly suited for deployment in real-world IIoT and edge environments. Unlike monolithic models that often overfit to specific data patterns, this approach leverages synergy between components: the transformer encodes long-term sequence behavior, the GAN balances class distributions, and the autoencoder denoises input to enhance anomaly sensitivity. Together, these elements form a resilient and versatile detection mechanism suitable for evolving industrial threats. While previous sections focused primarily on detection performance, this section extends the evaluation by addressing critical practical considerations: execution time, statistical robustness, and model consistency. These dimensions are vital for transitioning from lab-scale experimentation to industrial deployment, where constraints on computation, response time, and reliability are non-negotiable. In Edge or embedded IIoT systems, even a few seconds of additional delay or instability across runs can compromise the trustworthiness of a security framework.

We first examine the execution time required by each model. Efficiency is crucial in time-sensitive IIoT applications, especially in scenarios such as predictive maintenance, anomaly mitigation, or real-time threat response. A model that performs well but takes too long to converge or respond may not be viable in edge environments with limited computational resources. Next, we consider the results of a statistical t-test to determine whether the observed performance improvements of the proposed model are statistically significant. This analysis adds rigor to the evaluation, confirming that gains are not artifacts of randomness or dataset bias. Lastly, we assess variance across multiple experimental runs to evaluate the consistency and reliability of each model. High-performing systems must not only achieve strong average performance but must do so consistently under different initializations, random seeds, and training conditions. Low variance suggests robustness and reproducibility—two attributes that are essential for safety-critical systems, particularly in industrial automation, smart grids, or autonomous monitoring environments. These three additional dimensions together provide a holistic view of the practicality, deployability, and stability of Transformer–GAN–AE in real-world industrial cybersecurity ecosystems.

Table 4 presents a comparative runtime analysis of all evaluated models under different RMSE convergence thresholds on the WUSTL-IIoT-2021 dataset. The table reports the time (in seconds) each model requires to reach RMSE values below 15, 10, 5, and 2.5. This analysis offers insight into the efficiency and scalability of each algorithm when deployed in scenarios that demand varying levels of precision or convergence quality. It highlights not only the final performance but also the speed with which each model can achieve meaningful predictive accuracy. The Transformer–GAN–AE model exhibits a significant advantage in runtime efficiency across all thresholds. It reaches RMSE < 15 in just 15 s and converges to RMSE < 2.5 in only 128 s—outperforming all other models by a large margin. BERT, TST, and DBN show relatively moderate efficiency, but their inability to reach RMSE < 5 or 2.5 within reasonable time frames underlines the limitations of their training dynamics and optimization methods. In contrast, models such as CNN, GAN, and XGBoost fail to reach thresholds below RMSE < 10 altogether, indicating poor convergence behavior under higher precision demands.

Table 4. Runtime comparison of models at different RMSE thresholds on the WUSTL-IIoT-2021 dataset.

Proposed Methods	**Run Time (s)**			
	RMSE < 15	**RMSE < 10**	**RMSE < 5**	**RMSE < 2.5**
Transformer–GAN–AE	15	49	89	128
BERT	96	218	419	-
TST	110	231	-	-
DBN	125	286	-	-
GAN	163	341	-	-
CNN	180	-	-	-
XGboost	214	-	-	-

These findings reinforce the architectural and optimization benefits of the proposed framework. The use of the IChOA optimizer accelerates convergence and avoids stagnation in local minima. This contributes to reduced training time while still achieving high-quality results. From an industrial perspective, such rapid convergence is critical for real-time applications and retraining in Edge-based environments where resources and time are constrained. The results in Table 4 confirm that Transformer–GAN–AE not only performs better but also learns faster, making it a practical and deployable solution for mission-critical IIoT intrusion detection systems.

Analyzing computational complexity is critical for understanding the scalability and deployment feasibility of DL models, particularly in latency-sensitive and resource-constrained IIoT and edge environments. The proposed Transformer–GAN–AE framework is optimized by the IChOA consists of multiple interdependent modules, each contributing differently to the overall computational cost. The total complexity can be approximated by summing the complexities of its four core components: transformer encoder, GAN, AE, and IChOA. The transformer encoder, which captures long-range dependencies using multi-head self-attention, incurs the highest cost when sequence lengths are large. The complexity of a single forward pass is given by Equation (35):

$$O_{Transformer} = O(T \times A \times L^2 \times F) \tag{35}$$

where T is the number of transformer layers; A is the number of attention heads per layer; L is the sequence length; F is the feature dimensionality of each token. The GAN component consists of a generator and a discriminator, both assumed to be multilayer perceptrons (MLPs) with k hidden layers of width w. For a batch size of n, latent vector of dimension z, and output dimension D, the combined forward–backward complexity is estimated as Equation (36):

$$O_{GAN} = O(n \times (z \times w + k \times w^2 + w \times D)) \tag{36}$$

where n is the number of samples per batch; z is the size of the input noise vector; w is the width of each hidden layer; k is the number of layers in each network; and D is the dimensionality of the generated data sample. This formulation reflects the training cost of both generator and discriminator. The *AE*, consisting of symmetric encoder and decoder subnetworks, also relies on fully connected layers. Assuming input dimensionality d, encoder depth k_e, decoder depth k_d, and hidden width w, the total complexity is estimated as Equation (37):

$$O_{AE} = O(n \times (d \times w + (k_e + k_d) \times w^2 + w \times D)) \tag{37}$$

The encoder compresses the input into a latent space, and the decoder reconstructs it back to the original feature space. The *IChOA* optimizer introduces additional overhead by iteratively refining the model's hyperparameters. Let I be the number of iterations, P the population size, and D_h the number of hyperparameters being optimized. The optimization cost is estimated as Equation (38):

$$O_{IChOA} = O(P \times I \times D_h) \tag{38}$$

The total computational complexity of the proposed system can be approximated as the sum of the above components (Equation (39)):

$$O_{Total} = O_{Transformer} + O_{GAN} + O_{AE} + O_{IChOA} \tag{39}$$

This composite expression highlights the layered nature of the model's complexity, with Transformer dominating in long-sequence scenarios, *GAN* and *AE* influencing training runtime and memory usage, and *IChOA* contributing to overall optimization cost.

Table 5 presents the results of statistical significance testing using independent *t*-tests, comparing the Transformer–GAN–AE model against each of the six baseline models. The table includes the calculated *p*-values, the evaluation result at a confidence level of α = 0.01, and a binary assessment of whether the observed differences in performance are statistically significant. This analysis strengthens the empirical evidence by quantifying whether the improvements offered by Transformer–GAN–AE are meaningful and reproducible rather than resulting from random variation or sampling bias. The *t*-test is a widely used statistical method for comparing the means of two independent samples to determine if there is a statistically significant difference between them. In this context, each model's performance metric (such as accuracy or RMSE) is treated as a distribution over multiple experimental runs.

Table 5. Statistical significance analysis of the Transformer–GAN–AE compared to other models.

Model	Statistical *t*-Tests		
	***p*-Value**	**Results**	**α**
Transformer–GAN–AE vs. BERT	0.0005	Significant	0.01
Transformer–GAN–AE vs. TST	0.0002	Significant	0.01
Transformer–GAN–AE vs. DBN	0.00007	Significant	0.01
Transformer–GAN–AE vs. GAN	0.00004	Significant	0.01
Transformer–GAN–AE vs. CNN	0.00003	Significant	0.01
Transformer–GAN–AE vs. XGboost	0.00001	Significant	0.01

A *p*-value below the significance threshold (α = 0.01) indicates that the null hypothesis (i.e., there is no difference between the models) can be rejected, and the difference in performance is unlikely to be due to chance alone. This is critical for verifying that improvements are not only numerical but also statistically valid. As shown in Table 5, all *p*-values fall well below the 0.01 threshold, confirming that the performance gains achieved by Transformer–GAN–AE are statistically significant across all comparisons. The smallest *p*-values are observed against XGBoost and CNN (0.00001 and 0.00003, respectively), emphasizing the wide performance margin. Even in comparison with stronger models such as BERT and TST, the *p*-values remain highly significant (0.0005 and 0.0002). These results confirm that the superiority of the proposed model is not only consistent but also statistically validated, reinforcing its reliability and robustness as an advanced detection framework for real-world IIoT systems.

Table 6 reports the variance of each model's performance across repeated experimental trials on the WUSTL-IIoT-2021, Edge-IIoTset, and TON_IoT datasets. Variance quantifies how much the results fluctuate due to stochastic elements such as weight initialization, training data shuffling, or learning dynamics. Lower variance indicates greater consistency and robustness, critical qualities for models intended for deployment in sensitive industrial settings, where unpredictability in behavior could undermine trust, safety, and real-time responsiveness. As shown in Table 6, the proposed Transformer–GAN–AE achieves exceptionally low variance values on all datasets, 0.00009, 0.00011, and 0.00008, demonstrating its remarkable stability and reliability. In contrast, all baseline models exhibit higher levels of fluctuation, particularly CNN and XGBoost, whose variance reaches as high as 8.19 and 7.52, respectively. Even strong baselines such as BERT and TST show substantially higher variance compared to Transformer–GAN–AE, suggesting that while their average performance may be acceptable, they are more sensitive to random factors and less dependable under operational deployment. These findings underscore one of the key advantages of the proposed architecture and optimization strategy. The low variance achieved by Transformer–GAN–AE is primarily attributed to the integrated IChOA optimizer, particularly the balancer agent, which enhances convergence stability by adaptively balancing exploration and exploitation. This stability ensures that the model performs reliably even when retrained or deployed on new data slices, making it well-suited for environments where security responses must be both accurate and repeatable. In real-world IIoT systems (where false alarms or undetected intrusions can have serious implications), such consistency is not a luxury, but a necessity.

Table 6. Variance of model performance across multiple independent runs.

Algorithm	Variance		
	WUSTL-IIoT-2021	**Edge-IIoTset**	**TON_IoT**
Transformer-GAN-AE	0.00009	0.00011	0.00008
BERT	1.98652	2.31425	1.59632
TST	2.02413	3.54789	1.90258
DBN	2.14756	3.96321	3.41256
GAN	3.25058	4.09652	3.08652
CNN	5.21478	6.20035	4.98563
XGboost	8.32526	8.18963	7.52147

5. Conclusions

Intrusion detection in IIoT and edge environments remains a critical challenge due to the complexity, heterogeneity, and high dimensionality of networked systems. Traditional models often struggle to adapt to evolving threat patterns, class imbalance, and temporal dependencies present in real-world data. In this paper, we proposed a novel hybrid DL framework (Transformer-GAN-AE) designed to address these challenges through a synergistic combination of generative learning, sequence modeling, and reconstruction-based anomaly detection. The architecture integrates a transformer encoder to capture long-range temporal dependencies, a GAN to synthesize realistic minority class samples, and an AE to refine features and suppress noise in the data. To enhance model generalization and training stability, we incorporated an IChOA featuring a novel balancer agent that adaptively regulates the trade-off between exploration and exploitation during hyperparameter tuning. The proposed model was evaluated on three publicly available benchmark datasets, WUSTL-IIoT-2021, Edge-IIoTset, and TON_IoT, each representing diverse IIoT security scenarios.

Extensive experiments, including comparisons with state-of-the-art models such as CNN, DBN, TST, BERT, and XGBoost, demonstrated that Transformer–GAN–AE consistently outperforms all baselines in terms of accuracy, recall, AUC, convergence speed, runtime efficiency, and statistical robustness. On the WUSTL-IIoT-2021 dataset, the proposed model achieved an accuracy of 97.86%, a recall of 98.63%, and an AUC of 99.18%, surpassing the next-best model (BERT) by 6.5%, 5.7%, and 4.8%, respectively. On the more complex TON_IoT dataset, Transformer–GAN–AE achieved 98.92% accuracy, 99.52% recall, and an almost perfect AUC of 99.87%, outperforming all alternatives across all metrics. Furthermore, the radar and ROC plots illustrated consistent dominance across performance dimensions, confirming the model's superior generalization and anomaly sensitivity. Beyond classification performance, the proposed model also exhibited superior training dynamics and operational stability. It reached an RMSE below 2.5 in just 128 s, while all other models either failed to converge or required significantly more time. A *t*-test analysis showed that improvements over baselines were statistically significant ($p < 0.01$ for all comparisons), while variance analysis revealed minimal fluctuation across repeated runs, less than 0.0001 on all datasets, highlighting Transformer–GAN–AE's robustness and consistency. These results collectively validate the effectiveness of the proposed architecture and its IChOA-based optimization in delivering a scalable, accurate, and dependable solution for real-world IIoT intrusion detection.

The findings of this study underscore a fundamental insight: addressing complex cyber threats in edge and IIoT environments requires not only higher accuracy but also architectural adaptability, robust optimization, and systemic consistency. The superior performance of Transformer–GAN–AE is not solely due to its composite structure, but rather to the cohesive interaction between its components, each of which fulfills a specific, complementary function. The transformer encoder captures evolving patterns over time, the GAN enhances data diversity and combats imbalance, and the AE filters anomalies with sensitivity to reconstruction variance. This multi-perspective learning framework proves more effective than any single-model counterpart, highlighting the necessity of hybrid architectures for modern cyber-physical systems. A second key takeaway is the critical importance of optimization strategy in DL pipelines. While many studies overlook this aspect, our integration of the IChOA with a novel balancer agent demonstrates that carefully designed evolutionary tuning mechanisms can significantly impact both performance and efficiency. The IChOA optimizer ensured not only faster convergence but also lower variance and greater reproducibility, which are essential for deploying models in real-time, safety-critical environments. In this context, optimization is not just a support function; it is an enabler of practical, scalable security intelligence. Perhaps most importantly, this work provides a template for resilient and generalizable intrusion detection systems in high-stakes industrial settings. Unlike conventional approaches that focus narrowly on detection rates, our framework takes a system-wide view, from data irregularities and feature dependencies to runtime cost and statistical confidence. These contributions are not limited to the datasets studied here; they signal a pathway forward for designing AI-driven cybersecurity tools that can learn adaptively, operate reliably, and scale seamlessly across diverse industrial infrastructures.

Building on the promising results of this work, future research can explore extending the proposed Transformer–GAN–AE framework to support online and continual learning in dynamic industrial environments. Real-world IIoT systems are highly non-stationary, attack patterns evolve, device behaviors shift, and new data streams emerge over time. Developing mechanisms that allow the model to adapt incrementally without full retraining will enhance its responsiveness and longevity in production. Additionally, the framework can be enriched to support multi-source and multi-modal data, combining network traffic with system logs, telemetry, and contextual metadata to provide a more holistic and accurate picture of security events. From a deployment perspective, the next step involves evaluating the model under real-time operational constraints, such as limited computing power, memory availability, and energy consumption typical of edge devices. Porting the model into containerized microservices or embedded AI runtimes could enable integration into industrial control systems and smart gateways. Moreover, validating the system through field trials or industry-grade testbeds will provide critical insights into latency, robustness under attack, and interoperability with existing security frameworks. These efforts will bring the proposed solution closer to full-scale adoption in industrial domains such as energy grids, smart manufacturing, and autonomous infrastructure monitoring, where the need for reliable, adaptive, and intelligent intrusion detection is more critical than ever.

As part of another future work, we plan to extend the proposed Transformer–GAN–AE framework toward privacy-preserving architectures tailored for sensitive IIoT applications. This includes exploring federated learning to enable collaborative model training across edge nodes without transmitting raw data and incorporating differential privacy mechanisms into the GAN component to safeguard minority-class information. Such enhancements would improve the model's applicability in privacy-critical domains such as smart grids, industrial control systems, and healthcare IoT.

Author Contributions: Conceptualization, A.S., P.S.M. and M.K.; methodology, A.S., P.S.M. and M.K.; software, A.S.; validation, A.S. and M.K.; formal analysis, P.S.M.; investigation, A.S. and P.S.M.; resources, M.K.; data curation, A.S. and P.S.M.; writing—original draft preparation, A.S., P.S.M. and M.K.; writing—review and editing, A.S., P.S.M. and M.K.; visualization, A.S.; supervision, M.K.; project administration, M.K.; funding acquisition, M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

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

References

1. Ullah, I.; Adhikari, D.; Su, X.; Palmieri, F.; Wu, C.; Choi, C. Integration of Data Science with the Intelligent IoT (IIoT): Current Challenges and Future Perspectives. *Digit. Commun. Netw.* **2025**, *11*, 280–298. [CrossRef]
2. Hao, H.; Xu, C.; Zhang, W.; Yang, S.; Muntean, G.M. Task-Driven Priority-Aware Computation Offloading Using Deep Reinforcement Learning. *IEEE Trans. Wirel. Commun* **2025**. *Early Access*. [CrossRef]
3. Najafi, F.; Kaveh, M.; Mosavi, M.R.; Brighente, A.; Conti, M. EPUF: An Entropy-Derived Latency-Based DRAM Physical Unclonable Function for Lightweight Authentication in Internet of Things. *IEEE Trans. Mob. Comput.* **2025**, *24*, 2422–2436. [CrossRef]
4. Peruthambi, V.; Pandiri, L.; Kaulwar, P.K.; Koppolu, H.K.R.; Adusupalli, B.; Pamisetty, A. Big Data-Driven Predictive Maintenance for Industrial IoT (IIoT) Systems. *Metall. Mater. Eng.* **2025**, *31*, 21–30. [CrossRef]

5. Babayigit, B.; Abubaker, M. Industrial Internet of Things: A Review of Improvements over Traditional SCADA Systems for Industrial Automation. *IEEE Syst. J.* **2023**, *18*, 120–133. [CrossRef]
6. Wyrwa, J. A Review of the European Union Financial Instruments Supporting the Innovative Activity of Enterprises in the Context of Industry 4.0 in the Years 2021–2027. *Entrep. Sustain. Issues* **2020**, *8*, 1146. [CrossRef] [PubMed]
7. Hassan, W.H. Current Research on Internet of Things (IoT) Security: A Survey. *Comput. Netw.* **2019**, *148*, 283–294.
8. Miri, S.; Kaveh, M.; Shahhoseini, H.S.; Mosavi, M.R.; Aghapour, S. On the Security of 'An Ultra-Lightweight and Secure Scheme for Communications of Smart Meters and Neighborhood Gateways by Utilization of an ARM Cortex-M Microcontroller'. *IET Inf. Secur.* **2023**, *17*, 544–551. [CrossRef]
9. Rathee, G.; Iqbal, R.; Kerrache, C.A.; Song, H. TrustNextGen: Security Aspects of Trustworthy Next Generation Industrial Internet of Things (IIoT). *IEEE Internet Things J.* **2024**, *11*, 25568–25576. [CrossRef]
10. Ismail, S.; Dandan, S.; Qushou, A.A. Intrusion Detection in IoT and IIoT: Comparing Lightweight Machine Learning Techniques Using TON_IoT, WUSTL-IIOT-2021, and EdgeIIoTset Datasets. *IEEE Access* **2025**, *13*, 73468–73485. [CrossRef]
11. Cetintav, I.; Sandikkaya, M.T. A Review of Lightweight IoT Authentication Protocols from the Perspective of Security Requirements, Computation, Communication, and Hardware Costs. *IEEE Access* **2025**, *13*, 37703–37723. [CrossRef]
12. Kaveh, M.; Mosavi, M.R. A Lightweight Mutual Authentication for Smart Grid Neighborhood Area Network Communications Based on Physically Unclonable Function. *IEEE Syst. J.* **2020**, *14*, 4535–4544. [CrossRef]
13. Cirne, A.; Sousa, P.R.; Resende, J.S.; Antunes, L. Hardware Security for Internet of Things Identity Assurance. *IEEE Commun. Surv. Tutor.* **2024**, *26*, 1041–1079. [CrossRef]
14. Fan, C.I.; Lai, C.I.; Medhane, D.V. CAKE-PUF: A Collaborative Authentication and Key Exchange Protocol Based on Physically Unclonable Functions for Industrial Internet of Things. *IEEE Internet Things J.* **2024**, *11*, 39709–39720. [CrossRef]
15. Kaveh, M.; Mosavi, M.R.; Martin, D.; Aghapour, S. An Efficient Authentication Protocol for Smart Grid Communication Based on On-Chip-Error-Correcting Physical Unclonable Function. *Sustain. Energy Grids Netw.* **2023**, *36*, 101228. [CrossRef]
16. Jain, U. Secure, Light-Weight and Dynamic PUF-Based Mutual Device Authentication Mechanism in Industrial IoT Networks. *Secur. Priv.* **2024**, *7*, e388. [CrossRef]
17. Ma, S.; Wang, H.; Li, Z.; Zhang, Q. A Novel Approach for Estimating Performance of IIoT-Based Virtual Control Train Sets under DoS Attacks. *Secur. Commun. Netw.* **2022**, *2022*, 1781757. [CrossRef]
18. Joshi, S.; Crowther, K.; Robinson, J. Tradeoffs in Key Rotation Strategies for Industrial Internet of Things Devices and Firmware. *Appl. Sci.* **2024**, *14*, 9942. [CrossRef]
19. Kaveh, M.; Yan, Z.; Jantti, R. Secrecy Performance Analysis of RIS-Aided Smart Grid Communications. *IEEE Trans. Ind. Inform.* **2024**, *20*, 5415–5427. [CrossRef]
20. Ghadi, F.; Kaveh, M.; Wong, K.; Martin, D. Physical Layer Security Performance of Cooperative Dual-RIS-Aided V2V NOMA Communications. *IEEE Syst. J.* **2024**, *18*, 2074–2084. [CrossRef]
21. Du, R.; Zhen, L. Multiuser Physical Layer Security Mechanism in the Wireless Communication System of the IIoT. *Comput. Secur.* **2022**, *113*, 102559. [CrossRef]
22. Kaveh, M.; Ghadi, F.R.; Jantti, R.; Yan, Z. Secrecy Performance Analysis of Backscatter Communications with Side Information. *Sensors* **2023**, *23*, 8358. [CrossRef]
23. Tan, S.F.; Samsudin, A. Recent Technologies, Security Countermeasure and Ongoing Challenges of Industrial Internet of Things (IIoT): A Survey. *Sensors* **2021**, *21*, 6647. [CrossRef]
24. Nuaimi, M.; Fourati, L.C.; Hamed, B.B. Intelligent Approaches Toward Intrusion Detection Systems for Industrial Internet of Things: A Systematic Comprehensive Review. *J. Netw. Comput. Appl.* **2023**, *215*, 103637. [CrossRef]
25. Alsoufi, M.A.; Razak, S.; Siraj, M.M.; Nafea, I.; Ghaleb, F.A.; Saeed, F.; Nasser, M. Anomaly-Based Intrusion Detection Systems in IoT Using Deep Learning: A Systematic Literature Review. *Appl. Sci.* **2021**, *11*, 8383. [CrossRef]
26. Mathew, S.S.; Hayawi, K.; Dawit, N.A.; Taleb, I.; Trabelsi, Z. Integration of Blockchain and Collaborative Intrusion Detection for Secure Data Transactions in Industrial IoT: A Survey. *Clust. Comput.* **2022**, *25*, 4129–4149. [CrossRef]
27. Otoum, Y.; Nayak, A. AS-IDS: Anomaly and Signature Based IDS for the Internet of Things. *J. Netw. Syst. Manag.* **2021**, *29*, 23. [CrossRef]
28. Nandanwar, H.; Katarya, R. Deep Learning Enabled Intrusion Detection System for Industrial IoT Environment. *Expert Syst. Appl.* **2024**, *249*, 123808. [CrossRef]
29. Peng, Y.; Tan, A.; Wu, J.; Bi, Y. Hierarchical Edge Computing: A Novel Multi-Source Multi-Dimensional Data Anomaly Detection Scheme for Industrial Internet of Things. *IEEE Access* **2019**, *7*, 111257–111270. [CrossRef]
30. Srivastav, S.; Shukla, A.K.; Kumar, S.; Muhuri, P.K. HYRIDE: Hybrid and Robust Intrusion Detection Approach for Enhancing Cybersecurity in Industry 4.0. *Internet Things* **2025**, *30*, 101492. [CrossRef]
31. Kumar, P.; Mullick, S.; Das, R.; Nandi, A.; Banerjee, I. IoTForge Pro: A Security Testbed for Generating Intrusion Dataset for Industrial IoT. *IEEE Internet Things J.* **2025**, *12*, 8453–8460. [CrossRef]

32. Martins, I.; Cecílio, J.; Ferreira, P.M.; Oliveira, A. Comparative Analysis of Cybersecurity Datasets in Industrial Control Systems. In Proceedings of the 2024 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0 & IoT), Trento, Italy, 29–31 May 2024; pp. 440–445.
33. Ruiz-Villafranca, S.; Roldán-Gómez, J.; Carrillo-Mondejar, J.; Martinez, J.L.; Gañán, C.H. WFE-Tab: Overcoming Limitations of TabPFN in IIoT-MEC Environments with a Weighted Fusion Ensemble-TabPFN Model for Improved IDS Performance. *Future Gener. Comput. Syst.* **2025**, *166*, 107707. [CrossRef]
34. Ruiz-Villafranca, S.; Roldán-Gómez, J.; Gómez, J.M.C.; Carrillo-Mondéjar, J.; Martinez, J.L. A TabPFN-Based Intrusion Detection System for the Industrial Internet of Things. *J. Supercomput.* **2024**, *80*, 20080–20117. [CrossRef]
35. Hassini, K.; Khalis, S.; Habibi, O.; Chemmakha, M.; Lazaar, M. An End-to-End Learning Approach for Enhancing Intrusion Detection in Industrial-Internet of Things. *Knowl.-Based Syst.* **2024**, *294*, 111785. [CrossRef]
36. Alzubi, O.A.; Alzubi, J.A.; Qiqieh, I.; Al-Zoubi, A.M. An IoT Intrusion Detection Approach Based on Salp Swarm and Artificial Neural Network. *Int. J. Netw. Manag.* **2025**, *35*, e2296. [CrossRef]
37. Abou-Elasaad, M.M.; Sayed, S.G.; El-Dakroury, M.M. Securing the Future: Real-Time Intrusion Detection in IIoT Smart Grids through Innovative AI Solutions. *J. Cybersecur. Inf. Manag.* **2025**, *15*, 2.
38. Singh, G.; Sood, K.; Rajalakshmi, P.; Nguyen, D.D.N.; Xiang, Y. Evaluating Federated Learning Based Intrusion Detection Scheme for Next Generation Networks. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 4816–4829. [CrossRef]
39. Koppula, M.; Leo Joseph, L.M. LNKDSEA: Machine Learning Based IoT/IIoT Attack Detection Method. In Proceedings of the 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS), Hyderabad, India, 19–21 April 2023; pp. 655–662.
40. Khatami, S.S.; Shoeibi, M.; Oskouei, A.E.; Martín, D.; Dashliboroun, M.K. 5DGWO-GAN: A Novel Five-Dimensional Gray Wolf Optimizer for Generative Adversarial Network-Enabled Intrusion Detection in IoT Systems. *Comput. Mater. Contin.* **2025**, *82*, 1. [CrossRef]
41. Li, T.; Cao, Y.; Ye, Q.; Zhang, Y. Generative Adversarial Networks (GAN) Model for Dynamically Adjusted Weld Pool Image toward Human-Based Model Predictive Control (MPC). *J. Manuf. Process.* **2025**, *141*, 210–221. [CrossRef]
42. Liao, W.; Yang, K.; Fu, W.; Tan, C.; Chen, B.; Shan, Y. A Review: The Application of Generative Adversarial Network for Mechanical Fault Diagnosis. *Meas. Sci. Technol.* **2024**, *35*, 062002. [CrossRef]
43. Xie, R.; Wen, J.; Quitadamo, A.; Cheng, J.; Shi, X. A deep auto-encoder model for gene expression prediction. *BMC Genom.* **2017**, *18*, 39–49. [CrossRef]
44. Huang, J.; Liu, Y.; Yang, X.; Lv, Z.; Peng, K. Cointegration stacked autoencoder model based on stationary features reconstruction for non-stationary process monitoring. *Process Saf. Environ. Prot.* **2025**, *193*, 1287–1299. [CrossRef]
45. Dalla-Torre, H.; Gonzalez, L.; Mendoza-Revilla, J.; Lopez Carranza, N.; Grzywaczewski, A.H.; Oteri, F.; Pierrot, T. Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics. *Nat. Methods* **2025**, *22*, 287–297. [CrossRef]
46. Wang, C.; Chen, Y.; Zhang, S.; Zhang, Q. Stock Market Index Prediction Using Deep Transformer Model. *Expert Syst. Appl.* **2022**, *208*, 118128. [CrossRef]
47. Zhang, X.; Lin, M.; Hong, Y.; Xiao, H.; Chen, C.; Chen, H. MSFT: A Multi-Scale Feature-Based Transformer Model for Arrhythmia Classification. *Biomed. Signal Process. Control* **2025**, *100*, 106968. [CrossRef]
48. Zhao, J.; Wang, Z.; Wu, Y.; Burke, A.F. Predictive Pretrained Transformer (PPT) for Real-Time Battery Health Diagnostics. *Appl. Energy* **2025**, *377*, 124746. [CrossRef]
49. Xiao, Y.; Shao, H.; Wang, J.; Yan, S.; Liu, B. Bayesian Variational Transformer: A Generalizable Model for Rotating Machinery Fault Diagnosis. *Mech. Syst. Signal Process.* **2024**, *207*, 110936. [CrossRef]
50. Khishe, M.; Mosavi, M.R. Chimp optimization algorithm. *Expert Syst. Appl.* **2020**, *149*, 113338. [CrossRef]
51. Qian, L.; Khishe, M.; Huang, Y.; Mirjalili, S. SEB-ChOA: An improved chimp optimization algorithm using spiral exploitation behavior. *Neural Comput. Appl.* **2024**, *36*, 4763–4786. [CrossRef]
52. Hamza, M.F.; Modu, B.; Almutairi, S.Z. Integration of the Chimp Optimization Algorithm and Rule-Based Energy Management Strategy for Enhanced Microgrid Performance Considering Energy Trading Pattern. *Electronics* **2025**, *14*, 2037. [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.