symmetry

MDPI

Article

Hybrid Time Series Transformer–Deep Belief Network for Robust Anomaly Detection in Mobile Communication Networks

Anita Ershadi Oskouei [1], Mehrdad Kaveh [2], Francisco Hernando-Gallego [3] and Diego Martín [2,*]

[1] School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ 07030, USA; aershadi@stevens.edu
[2] Department of Computer Science, Escuela de Ingeniería Informática de Segovia, Universidad de Valladolid, 40005 Segovia, Spain; mehrdadkaveh11@gmail.com
[3] Department of Applied Mathematics, Escuela de Ingeniería Informática de Segovia, Universidad de Valladolid, 40005 Segovia, Spain; fhernando@uva.es
* Correspondence: diego.martin.andres@uva.es

Abstract

The rapid evolution of 5G and emerging 6G networks has increased system complexity, data volume, and security risks, making anomaly detection vital for ensuring reliability and resilience. However, existing machine learning (ML)-based approaches still face challenges related to poor generalization, weak temporal modeling, and degraded accuracy under heterogeneous and imbalanced real-world conditions. To overcome these limitations, a hybrid time series transformer–deep belief network (HTST-DBN) is introduced, integrating the sequential modeling strength of TST with the hierarchical feature representation of DBN, while an improved orchard algorithm (IOA) performs adaptive hyper-parameter optimization. The framework also embodies the concept of symmetry and asymmetry. The IOA introduces controlled symmetry-breaking between exploration and exploitation, while the TST captures symmetric temporal patterns in network traffic whose asymmetric deviations often indicate anomalies. The proposed method is evaluated across four benchmark datasets (ToN-IoT, 5G-NIDD, CICDDoS2019, and Edge-IoTset) that capture diverse network environments, including 5G core traffic, IoT telemetry, mobile edge computing, and DDoS attacks. Experimental evaluation is conducted by benchmarking HTST-DBN against several state-of-the-art models, including TST, bidirectional encoder representations from transformers (BERT), DBN, deep reinforcement learning (DRL), convolutional neural network (CNN), and random forest (RF) classifiers. The proposed HTST-DBN achieves outstanding performance, with the highest accuracy reaching 99.61%, alongside strong recall and area under the curve (AUC) scores. The HTST-DBN framework presents a scalable and reliable solution for anomaly detection in next-generation mobile networks. Its hybrid architecture, reinforced by hyper-parameter optimization, enables effective learning in complex, dynamic, and heterogeneous environments, making it suitable for real-world deployment in future 5G/6G infrastructures.

Keywords: mobile communication networks; anomaly detection; 6G; time series transformer; deep belief network; improved orchard algorithm

Academic Editor: Abdelaziz Lberni

Received: 22 September 2025
Revised: 16 October 2025
Accepted: 19 October 2025
Published: 25 October 2025

Citation: Oskouei, A.E.; Kaveh, M.; Hernando-Gallego, F.; Martín, D. Hybrid Time Series Transformer–Deep Belief Network for Robust Anomaly Detection in Mobile Communication Networks. *Symmetry* **2025**, *17*, 1800. https://doi.org/10.3390/sym17111800

1. Introduction

The rapid proliferation of the Internet of Things (IoT) has transformed diverse sectors by enabling massive interconnectivity between sensors, actuators, and intelligent devices [1]. IoT applications span from smart grids, where real-time monitoring and

demand-response mechanisms improve energy efficiency [2] to healthcare systems that leverage wearable devices and remote patient monitoring for enhanced diagnosis and treatment [3]. Emerging paradigms such as batteryless IoT devices and backscatter communication have further extended the scope of ultra-low-power connectivity [4–7], supporting sustainable deployments in resource-constrained environments [8]. Meanwhile, intelligent transportation relies on vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication to enhance traffic safety and efficiency [9–11]. In contrast, industrial IoT (IIoT) facilitates predictive maintenance, automation, and process optimization [12]. Collectively, these advancements highlight IoT as a cornerstone of modern digital ecosystems and a key enabler of next-generation mobile communication networks [13].

Despite its transformative potential, IoT remains highly vulnerable to security threats due to its heterogeneous architecture, constrained resources, and massive attack surface [14–16]. Security challenges manifest across multiple layers, ranging from hardware vulnerabilities and side-channel attacks [17–19] to physical-layer risks such as jamming, spoofing, and eavesdropping [20–22]. At the network layer, issues such as routing manipulation, denial-of-service, and replay attacks compromise availability and data integrity, while lightweight authentication remains a persistent challenge in ensuring trust among resource-limited devices [23–26]. In this context, intrusion detection systems (IDSs) have become a critical line of defense for mobile IoT environments [27]. An IDS is tasked with monitoring network traffic, detecting abnormal behaviors, and identifying potential attacks in real-time, thereby safeguarding system reliability and continuity [28]. The integration of effective IDS mechanisms into mobile IoT is particularly vital, as these networks support critical infrastructures and latency-sensitive services that cannot tolerate prolonged downtime or undetected intrusions [29].

Machine learning (ML) and deep learning (DL) have emerged as powerful tools to address the limitations of traditional security mechanisms in IoT-based mobile networks [30–32]. Unlike signature-based approaches, which struggle against zero-day attacks or highly dynamic traffic, ML- and DL-driven models can autonomously learn complex patterns from heterogeneous and high-dimensional data, enabling robust generalization across diverse environments [33–35]. DL architectures, in particular, offer advanced temporal modeling and hierarchical feature abstraction, making them well-suited for capturing subtle anomalies hidden within noisy or imbalanced datasets. Their ability to adapt in real-time and -scale with massive traffic flows positions them as indispensable for next-generation IDS solutions in IoT, where resilience, accuracy, and efficiency are essential [35–37].

1.1. Related Works

In [30], an attention-based convolutional neural network (ABCNN) was introduced for intrusion detection in IoT networks. The approach integrated an attention mechanism to address minority classes and applied mutual information for feature selection. Evaluation on multiple IoT datasets, including Edge-IoTset, IoTID20, ToN-IoT, and CIC-IDS2017, against several ML/DL baselines demonstrated superior performance, with accuracy of 99.81%, precision of 98.02%, recall of 98.18%, and f1-score of 98.08%. In [31], an IDS for IoT networks was developed using ML-based feature selection combined with a stacked ensemble of classifiers. By applying the K-Best algorithm to select the 15 most relevant features, the method achieved accuracy up to 99.92%, significantly outperforming individual models.

The work [32] proposed a DL-based IDS for DDoS detection in IoT, focusing on extensive preprocessing (duplication removal, feature elimination, normalization) on the CIC-DDoS2019 dataset. Their model achieved 99.99% accuracy for binary classification and 99.30% accuracy for multiclass classification. They also reported inference time favorable compared to baseline models with fewer trainable parameters. The work [33] proposed

a DL-based IDS for DDoS detection using the sensors dataset. They evaluated models including Bi-long short-term memory (LSTM) and several ML baselines on CICDDoS2019, among others. The study reported that Bi-LSTM outperformed all other tested models. They achieved precision of 99.885% and accuracy of nearly 99.90%, with very low false alarm rates.

The work [34] investigated a network intrusion detection model that extended a radial basis function neural network (RBFNN) by embedding it as the policy network in an offline reinforcement learning (RF) framework. They learned all RBF parameters and weights end-to-end by gradient descent and explored the impact of using denser hidden-layers and varying the number of RBF kernels. Evaluation was performed on five datasets (NSL-KDD, UNSW-NB15, AWID, CICIDS2017, CICDDOS2019) under noisy and unbalanced conditions. The results showed that this extended RBF-NN achieved better performance metrics than state-of-the-art models, particularly strong accuracy and robustness to dataset imbalance. The work [35] compared feature selection and feature extraction techniques within ML-based intrusion detection systems for IoT networks. It applied methods such as Principal component analysis (PCA) and mutual information on datasets like ToN-IoT, Edge-IoTset, and CIC-DDoS2019. The study found that feature selection yielded higher accuracy, reaching about 99.90%, while also reducing model complexity and improving processing time.

The approach [36] developed DeepTransIDS, a transformer-based DL model for detecting DDoS attacks on the 5G-NIDD dataset. It utilized transformer encoder layers with self-attention to capture long-range dependencies in 5G traffic. The model was tested exclusively on the 5G-NIDD dataset and compared against CNN, LSTM, and ML baselines. It achieved high accuracy (≈99.94%) and recall (≈99.91%), outperforming other published models on the same dataset. The work [37] investigated DTL-5G, a deep transfer learning-based approach for DDoS attack detection in 5G and beyond networks. It leveraged pretrained DL models (such as CNN, BiLSTM, ResNet, and Inception) trained on a source domain, then fine-tuned them on the 5G-NIDD target dataset. They evaluated the approach under domain shift (limited labeled traffic in 5G-NIDD) and showed that transfer learning significantly improved detection compared to training from scratch. The reported performance included accuracy around 98.74%, demonstrating that DTL-5G was able to maintain high detection rates even in challenging settings.

The approach [38] evaluated several ML and hybrid models for DDoS detection in IoT-driven 6G energy hub networks. They compared algorithms like random forest (RF), gradient boosting (GB), support vector machine (SVM), decision tree (DT), and K-nearest neighbors (KNN), plus hybrids (RF + KNN, GB + KNN, GB + DT), on CICDDoS2019, KDD-CUP, and UNSW-NB datasets with different training size splits. They found that RF + KNN achieved highest accuracy (~99.44%) and that GB + DT had superior precision on some datasets, while GB + KNN attained the best recall under certain conditions. The work [39] investigated enhanced intrusion detection with advanced deep features and ensemble classifier techniques, where deep feature extraction via CNN, recurrent neural network (RNN), and autoencoder (AE) was combined with ensemble classifiers (RF + SVM) using soft voting. They evaluated the approach on the NSL-KDD and CICDDoS2019 datasets. The ensemble model with CNN-based features achieved near-perfect accuracy values (99.99% on NSL-KDD and 99.91% on CICDDoS2019), while the RNN- and autoencoder-based variants delivered ~99.30% accuracy on CICDDoS2019.

The work [40] investigated the weighted SVM kernel combined with adolescent identity search plus RF ensemble (wSVMAS-RF) for DDoS detection and categorization. It processed datasets including CICDDoS2019 and CICDoS2017 with a preprocessing pipeline using interval-reduced kernel PCA (IRKPCA) for dimensionality reduction. The method

achieved accuracy of about 99.74% on CICDDoS2019 and ~99.00% on CICDoS2017, along with competitive precision, recall, and low false positive rates. The work [41] investigated a one-dimensional CNN (OS-CNN) model for network intrusion detection. They implemented four temporal-convolution approaches, such as time series transformer (TST), and compared them on multiple intrusion detection datasets, including CICDDoS2019 and UNSW-NB15. They demonstrated that TST and OS-CNN achieved the highest accuracy, both exceeding 98.5% accuracy, and showed superior area under the curve (AUC) performance compared to baseline CNN and RNN models.

The work [42] investigated a stacking ensemble approach for DDoS attack detection in software-defined cyber-physical systems (SD-CPS), relying on DL base models. They evaluated their method on the SDN-specific dataset and CICDDoS2019, including both binary and multiclass scenarios. The method achieved accuracy above 99% in both binary and multiclass classifications, maintaining robustness even with unknown traffic distribution. The work [43] proposed 5G-SIID, a hybrid DDoS intrusion detection system for 5G IoT networks combining deep neural network (DNN) and classical ML classifiers (RF, DT, SVM, KNN). It used PCA for feature selection, SMOTE for imbalance correction, and standard scaling, and was evaluated on Edge-IIoTset, among others. The DNN achieved 100% accuracy for binary classification, ~96.15% for 6-class classification, and ~94.68% for 15-class classification.

The study [44] investigated a complete pipeline integrating exploratory data analysis (EDA) with DL for intrusion detection in softwarized 5G networks. It applied preprocessing steps such as feature correlation, scaling, and class balancing, followed by DL classifiers including CNN and LSTM. The approach was validated on the 5G-NIDD dataset and achieved accuracy exceeding 99%, with strong precision and recall, demonstrating the effectiveness of coupling EDA with DL for 5G intrusion detection. The work [45] developed a hybrid feature-selection approach to anomaly intrusion detection in IoT networks, combining filter and wrapper methods. It used SMOTE to address class imbalance and applied classifiers such as DT, RF, and KNN. The method was tested on the BoT-IoT, ToN-IoT, and CIC-DDoS2019 datasets. The DT classifier achieved accuracy between 99.82 and 100%, with detection times between 0.02 and 0.15 s, demonstrating both high efficiency and fast response.

1.2. Paper Motivation, Contribution, and Organization

Existing research has advanced IoT intrusion detection through various ML and DL models, ranging from attention-based CNNs [30] to ensemble classifiers [31], transformer-based models [36], and hybrid deep feature extractors [39]. While these approaches achieved high accuracy on individual datasets, they often struggled to generalize across heterogeneous environments. Many of them were evaluated under constrained conditions or relied on limited datasets, raising concerns about robustness in large-scale, multi-source mobile communication networks. Moreover, several solutions lacked the mechanisms to effectively capture both temporal dependencies in traffic and deep hierarchical representations, which are critical for detecting subtle anomalies in noisy or imbalanced IoT traffic.

In addition, most prior works have treated anomaly detection either as a pure classification problem or as a temporal sequence modeling task, without sufficiently bridging the gap between these two perspectives. This separation has often led to architectures that either excel in learning global sequential patterns, but which overlook fine-grained features or capture local abstractions while failing to model long-range dependencies. Furthermore, the lack of adaptive optimization strategies has limited their scalability when deployed in dynamic environments such as 5G and beyond.

Another notable limitation is the challenge of balancing exploration and exploitation in optimization when tuning complex DL architectures. Existing methods frequently use grid search or simple heuristics, which are computationally expensive and prone to suboptimal convergence. Even works leveraging meta-heuristics [40] tend to emphasize either exploration or exploitation, limiting adaptability in dynamic and high-dimensional scenarios like 5G/6G networks. Similarly, while some approaches introduce transfer learning [37] or feature selection [35,45], they still lack integrated optimization frameworks capable of adaptively fine-tuning hyperparameters to sustain performance across diverse datasets such as CICDDoS2019, ToN-IoT, Edge-IIoTset, and 5G-NIDD. To address these gaps, this paper proposes the HTST-DBN framework, which integrates the sequence-modeling strength of the TST architecture with the deep feature abstraction of the deep belief network (DBN). The architecture is optimized using an improved orchard algorithm (IOA) enhanced by a seasonal migration operator, designed to balance exploration and exploitation more effectively than existing methods.

In the context of mobile communication networks, normal traffic flows often exhibit symmetric temporal or statistical patterns, whereas anomalies manifest as asymmetric deviations caused by attacks, failures, or dynamic environmental factors. Within the proposed HTST-DBN framework, the TST models such temporal symmetries through self-attention, learning recurring and balanced dependencies across sequences, while the DBN captures higher-level mirrored representations that enhance robustness against noise and imbalance. Together, these components enable the model to recognize when expected symmetries in network behavior are disrupted. At the optimization level, the IOA introduces deliberate asymmetry through its seasonal-migration operator. By alternating probabilistically between convergence toward the global best and divergence toward dissimilar candidates, the IOA performs controlled symmetry-breaking in the search space, maintaining equilibrium between exploration and exploitation. This mechanism parallels the balance between structural symmetry and purposeful asymmetry found in circuit and system design. In that broader engineering sense, the same principles underlying the HTST-DBN and IOA can inspire applications in electronic design automation (EDA). Hence, the framework embodies a unifying view in which detecting or exploiting asymmetry becomes a fundamental tool for achieving optimality across both communication-network intelligence and design-automation domains. Therefore, the main contributions of this paper are as follows:

- A novel anomaly detection framework (HTST-DBN) is proposed for next-generation mobile networks, addressing the challenges of noisy, imbalanced, and heterogeneous IoT traffic through adaptive modeling and optimization.
- The proposed method combines a TST for temporal dependency learning, a DBN for hierarchical feature abstraction, and an IOA for adaptive optimization. Their integration within a single pipeline allows the model to capture sequential patterns, extract deep latent features, and fine-tune hyper-parameters adaptively, resulting in better generalization and robustness compared to standalone approaches.
- A novel IOA with a seasonal migration operator is introduced, which adaptively balances exploration and exploitation. This mechanism prevents premature convergence, ensures diverse search, and refines promising regions more effectively, leading to stronger and more reliable hyper-parameter optimization.
- The proposed model is extensively benchmarked against state-of-the-art algorithms, including BERT, deep reinforcement learning (DRL), CNN, and RF, across four diverse datasets (ToN-IoT, 5G-NIDD, CICDDoS2019, and Edge-IoTset), demonstrating superior performance and stability.

The remainder of this paper is organized as follows: Section 2 presents the overall research workflow, including the benchmark datasets, data preprocessing pipeline, and the proposed HTST-DBN methodology with its core components (TST, DBN, and IOA). Section 3 reports the experimental setup and performance evaluation, providing comparative results against state-of-the-art models. Section 4 offers a detailed discussion, highlighting robustness, statistical significance, and real-world applicability. Finally, Section 5 concludes the paper with key findings, and outlines promising directions for future research.

2. Materials and Proposed Methods

Figure 1 provides a high-level overview of the research methodology adopted in this work. Four benchmark datasets (ToN-IoT, 5G-NIDD, CICDDoS2019, and Edge-IoTset) serve as the foundation for model development and evaluation. The raw data streams undergo a preparation stage consisting of imputation, normalization, feature selection, and label encoding to ensure quality, consistency, and suitability for subsequent learning. Once preprocessed, the datasets are introduced into the proposed hybrid framework, which integrates a time series transformer (TST) with a deep belief network (DBN), further enhanced by the improved orchard algorithm (IOA) for adaptive optimization.

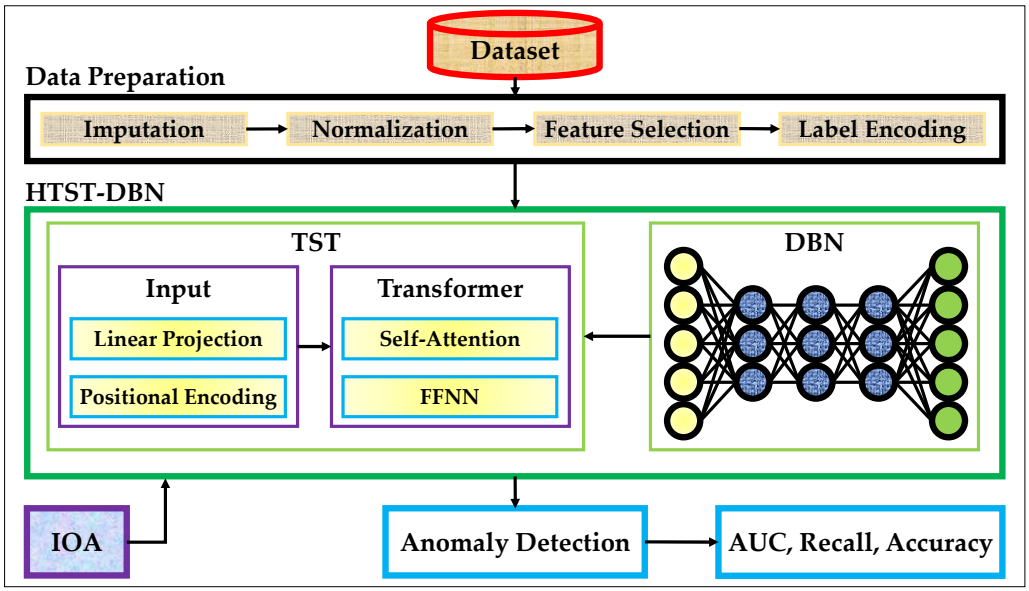

Figure 1. Research methodology framework.

As shown in the figure, the proposed HTST-DBN pipeline follows a structured flow from raw data to anomaly detection outcomes. The TST captures temporal dependencies in the input, the DBN abstracts high-level features, and IOA optimizes the architecture and hyper-parameters. The outputs of the hybrid model are evaluated through standard classification metrics such as accuracy, precision, recall, and area under the curve (AUC). This end-to-end design enables effective detection of anomalies in heterogeneous and dynamic mobile communication environments, while the detailed mechanisms of each component are elaborated in the subsequent subsections.

2.1. Dataset

In this study, four benchmark datasets were employed to comprehensively evaluate the proposed HTST-DBN framework [30,37,38]. These datasets collectively represent diverse environments, including IoT telemetry, 5G core traffic, DDoS attacks, and edge-enabled IoT systems. Their heterogeneity in traffic sources, feature dimensions, and class

distributions provides a rigorous testing ground for anomaly detection models. Validation across all four datasets demonstrates that the proposed method remains accurate under controlled conditions and robust to noise, imbalance, and variability inherent in real-world communication networks [37–39].

The ToN-IoT dataset is derived from a large-scale testbed for IoT and network systems [30,35]. It integrates telemetry from IoT devices, network traffic, and operating system logs, offering multi-source data streams. The dataset contains both benign and attack classes, including DoS, DDoS, backdoor, injection, ransomware, and password attacks. Its diversity makes it particularly valuable for validating the generalization ability of anomaly detection frameworks across heterogeneous input modalities. The 5G-NIDD dataset focuses on non-IP data delivery in 5G core networks, reflecting traffic patterns generated by IoT devices that operate without traditional IP stacks. This dataset captures signaling behavior, session establishment, and control-plane data flows in a 5G environment. The primary classes include normal device communications and various anomalous behaviors such as misconfigurations, flooding, and signaling abuse. Because it represents next-generation cellular networks, it is essential for evaluating anomaly detection models intended for 5G and beyond [36,37].

The CICDDoS2019 dataset is a well-established benchmark for DDoS detection. It was generated using a realistic testbed by simulating a wide range of DDoS attack vectors, including UDP floods, TCP SYN floods, HTTP floods, and amplification attacks, alongside normal traffic [38–40]. The dataset includes flow-based features such as packet counts, byte counts, and inter-arrival times. Its fine-grained labeling of different DDoS categories allows anomaly detection models to be tested not only on binary classification but also on multiclass detection of attack types. The Edge-IoTset dataset is designed for anomaly detection in edge-enabled IoT scenarios. It contains traffic traces from IoT devices connected through edge gateways, reflecting realistic deployments where computational resources and network conditions vary. The dataset includes both benign and malicious traffic, with attack types such as botnet, brute force, and reconnaissance. Its parameters cover packet-level and session-level features, making it highly representative of IoT devices [30,35].

To ensure the quality and consistency of the datasets used in this study, a systematic data preparation process was applied before feeding them into the HTST-DBN framework. Each dataset contains heterogeneous features and varying levels of noise, imbalance, and missing entries. Without careful preprocessing, these irregularities could severely hinder the performance of the model. Accordingly, a four-step preprocessing pipeline was designed, consisting of imputation, normalization, feature selection, and label encoding, to transform the raw data into a standardized and learning-ready format. All operations were implemented in Python 3.10 using Scikit-learn 1.2 and Imbalanced-learn 0.10 libraries, ensuring methodological transparency and cross-dataset consistency.

Incomplete or missing entries, especially in the ToN-IoT and Edge-IoTset datasets caused by packet loss or irregular telemetry sampling, were corrected using the KNN imputation algorithm with $K = 5$. For each missing value, the five most similar samples were identified in Euclidean space, and their mean value was used for replacement. This method preserves the non-linear relationships among correlated features and maintains the local distribution structure of network traffic, avoiding the statistical bias often introduced by mean or median imputation. The resulting datasets retained full record integrity while minimizing variance distortion across continuous attributes. Because the selected benchmarks contain features with highly diverse numeric scales (from packet counts in the hundreds to byte rates exceeding one million) all numerical attributes were normalized using Min–Max scaling ($x' = \frac{x - x_{min}}{x_{max} - x_{min}}$) into the interval $[0, 1]$. This scaling stabilizes gradient-based optimization, prevents domination of large-magnitude variables, and accel-

erates convergence of both the Transformer and DBN components. The Min–Max approach was favored over Z-score normalization because it preserves sparsity and boundedness, which are critical for time series modeling stability in traffic data.

To reduce dimensionality and improve generalization, a two-stage feature-selection process was employed. First, features with near-zero variance ($<10^{-3}$) were removed to eliminate non-informative attributes. Second, mutual information (MI) analysis was performed to quantify the dependency between each remaining feature and its corresponding label. Features were ranked according to MI scores, and the top-K attributes were retained ($K = 25 - 40$ depending on dataset dimensionality). This hybrid filter approach effectively discarded redundant variables while preserving those most strongly correlated with the target classes. The dimensional statistics before and after selection for each dataset are summarized in Table 1. The reduction percentage ($\left(1 - \frac{Selected\ Features}{Initial\ Features}\right) \times 100$) quantifies the extent to which redundant or low-variance features were eliminated, demonstrating the efficiency of the mutual-information-based selection process. Across all datasets, the feature space was reduced by roughly half, highlighting a strong balance between dimensional compression and information retention. This effective reduction confirms that the preprocessing pipeline successfully minimized noise while preserving the most discriminative attributes for anomaly detection.

Table 1. Feature statistics before and after selection.

Dataset	Initial Features	Selected Features	Reduction (%)
ToN-IoT	55	25	54.5
5G-NIDD	43	20	53.4
CICDDoS2019	88	40	54.5
Edge-IoTset	46	22	52.1

Severe class imbalance (particularly evident in ToN-IoT and Edge-IoTset) was mitigated using the synthetic minority over-sampling technique (SMOTE). SMOTE generates synthetic minority examples by interpolating between neighboring samples in feature space, equalizing class distributions and preventing bias toward majority categories. This oversampling was applied solely to the training subset to avoid information leakage. Finally, all categorical attack labels (e.g., normal, DoS, DDoS, botnet, brute-force) were numerically encoded via LabelEncoder to establish a consistent label space for supervised learning and evaluation. Through this structured preprocessing pipeline, all datasets were standardized, balanced, and dimensionally optimized, forming a stable and reproducible foundation for subsequent modeling with the proposed HTST-DBN architecture.

2.2. TST

The TST is a specialized adaptation of the Transformer architecture designed to address the unique characteristics of sequential temporal data. While standard Transformers such as vision Transformer have been widely adopted in natural language processing and computer vision, they are not inherently optimized for time-dependent structures [46]. TST introduces a tailored processing pipeline consisting of linear projection, positional encoding, and multi-head self-attention to capture long-range temporal dependencies in sequential inputs. Unlike recurrent neural networks (RNNs) or convolutional neural networks (CNNs), which often struggle with vanishing gradients or limited receptive fields, TST provides a non-recursive mechanism capable of modeling complex temporal correlations over extended horizons. For anomaly detection in mobile communication networks, where traffic traces and telemetry streams naturally follow sequential patterns, the strengths of TST become particularly valuable [47]. Conventional ML or even general-purpose Transformer models often fail to capture noisy, sparse, and highly dynamic temporal structures across

heterogeneous environments such as 5G cores, IoT telemetry, and mobile edge systems. TST addresses these limitations by (i) effectively learning long-term temporal dependencies, (ii) providing robustness against irregular and imbalanced data, and (iii) offering better generalization across multi-source and edge-enabled scenarios. These advantages make TST a crucial component of our hybrid framework, enabling reliable representation learning prior to deep feature abstraction with DBN [48].

The overall structure of TST is illustrated in Figure 2, which depicts the transformation pipeline from raw input series to anomaly scores. The input time series is first mapped into a fixed-dimensional space through linear projection, after which positional encoding enriches the embeddings with temporal order information. These encoded sequences are then processed by the Transformer encoder, where the multi-head self-attention mechanism evaluates contextual relationships among elements by computing query–key–value (Q, K, V) interactions through scaled dot-product attention. The outputs are normalized, passed through feed-forward layers, and aggregated into final embeddings that form the output representation. From these embeddings, an anomaly score is derived, reflecting the model's ability to highlight irregular or malicious behaviors in network traffic. This modular architecture underpins the strength of TST in modeling sequential patterns critical for anomaly detection [46–48].

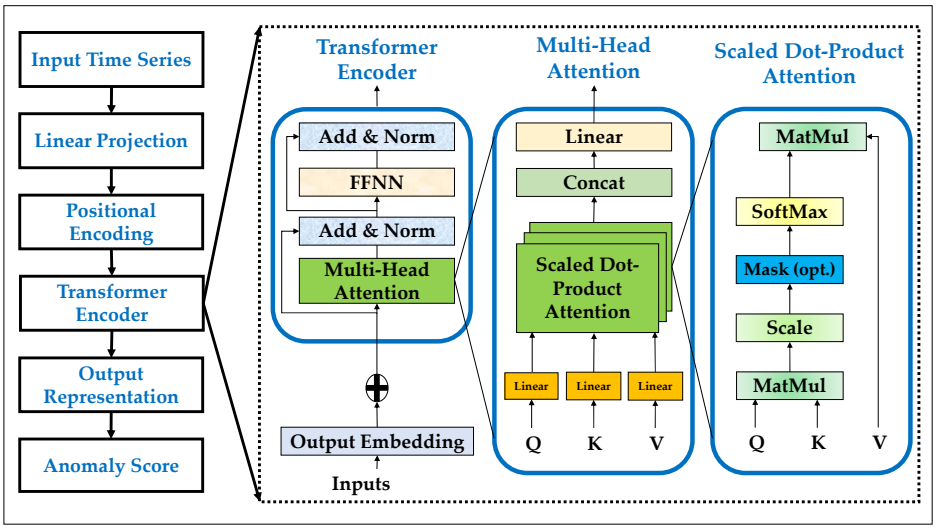

Figure 2. The basic TST architecture.

Equation (1) represents the mapping of a raw time series $x \in \mathbb{R}^{l/d_{in}}$ of length L and input dimension d_{in} into a latent embedding space of dimension d_{model}:

$$z_0 = xW_{proj} + b_{proj} \tag{1}$$

Here, $W_{proj} \in \mathbb{R}^{d_{in} \times d_{model}}$ is the projection matrix; $b_{proj} \in \mathbb{R}^{d_{model}}$ is the bias vector; $z_0 \in \mathbb{R}^{L \times d_{model}}$ denotes the projected embeddings. Since the attention mechanism itself is permutation-invariant, sequential order is injected using positional encoding as shown in Equation (2) [46]:

$$z_0' = z_0 + P \tag{2}$$

where $P \in \mathbb{R}^{L \times d_{model}}$ is the positional encoding matrix constructed from sinusoidal functions of varying frequencies. Once the sequence is embedded, queries, keys, and values are obtained via linear transformations, as defined in Equation (3):

$$Q = z_0' W^Q, \quad K = z_0' W^K, \quad V = z_0' W^V \tag{3}$$

Here, W^Q, W^K, $W^V \in \mathbb{R}^{d_k \times d_{model}}$ are the trainable weight matrices; d_k is the dimension of the projected query–key space. These components feed into the scaled dot-product attention, which is at the core of the self-attention mechanism. As shown in Equation (4), attention is computed by normalizing similarity scores between queries and keys and then weighting the values accordingly [47]:

$$Attention\ (Q,\ K,\ V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \tag{4}$$

In practice, multiple attention heads are used to capture diverse temporal dependencies. Equation (5) defines multi-head attention (MHA):

$$MHA\ (Q,\ K,\ V) = Concat(H_1,\ H_2,\ \ldots,\ H_h)W^O \tag{5}$$

Here, $H_i \in \mathbb{R}^{d_k \times d_{model}}$ is an independent attention output, and W^O projects the concatenated outputs back into the model dimension. The result is passed through a position-wise feed-forward network (FFN) as defined in Equation (6):

$$FFN\ (z) = \max(0,\ zW_1,\ + b_1)W_2 + b_2 \tag{6}$$

where W is the weight matrix; b is bias vector. This non-linear transformation further enriches the learned representation. To stabilize training and preserve gradient flow, residual connections combined with layer normalization are applied after each sublayer, as captured in Equation (7):

$$z' = LayerNorm(z + \text{Sublayer}(z)) \tag{7}$$

where $\text{Sublayer}(.)$ denotes either the multi-head attention block or the FFN. By stacking N such encoder layers, the model produces the final sequence representation, as described in Equation (8):

$$Z = Encoder(z_0') \tag{8}$$

Here, $Z \in \mathbb{R}^{L \times d_{model}}$ captures the contextual embeddings of the time series across all layers.

Finally, anomaly detection requires mapping these embeddings into a task-specific score, as formulated in Equation (9) [48]:

$$s = f(Z) \tag{9}$$

where $f(.)$ may represent a classification head or a reconstruction error metric, depending on whether the task is supervised or unsupervised.

Recent research has further advanced attention-based frameworks by integrating physical constraints and structural optimization to enhance interpretability, robustness, and computational efficiency. These studies demonstrate that attention mechanisms can effectively generalize across diverse time series and perception tasks, confirming their suitability for real-world IoT and intelligent monitoring applications [49,50]. Such findings further justify the adoption of the TST architecture in the proposed hybrid framework.

2.3. DBN

DBN was first introduced by Hinton and colleagues as a generative probabilistic model composed of multiple layers of stochastic latent variables. A DBN is built by stacking restricted Boltzmann machines (RBMs), where each RBM learns a hierarchical representation of the input data in an unsupervised manner [51]. Through greedy layer-wise pretrain-

ing followed by supervised fine-tuning, DBNs can capture increasingly abstract features from raw data. This hierarchical structure enables DBNs to address challenges such as non-linearity, high dimensionality, and limited labeled data, which are common in many real-world scenarios. By combining generative learning with discriminative fine-tuning, DBNs have been successfully applied in diverse domains such as speech recognition, image classification, and network traffic modeling. The primary advantages of DBNs lie in their ability to model complex probability distributions and to extract deep feature representations without requiring large-scale labeled datasets. Compared with shallow models or traditional machine learning techniques, DBNs demonstrate stronger generalization, robustness to noise, and improved performance on imbalanced or heterogeneous datasets. These properties make them particularly suitable for network anomaly detection, where traffic traces are often high-dimensional, noisy, and partially labeled [52].

Figure 3 illustrates the structure of a DBN, where the network is formed by sequentially stacking RBMs. The architecture begins with an input layer that feeds raw data into the first RBM. Each RBM consists of a visible layer and a hidden layer, where the hidden activations from one RBM serve as the visible inputs for the next, progressively forming deeper feature abstractions. The figure shows three hidden layers, highlighting how low-level features extracted in the early layers are transformed into increasingly complex and abstract representations at deeper layers. Finally, the output layer is connected to the last hidden layer to support supervised tasks such as classification. This hierarchical organization allows DBNs to capture both local patterns and high-level dependencies in the data. Through unsupervised pretraining of each RBM, the model learns robust feature representations even in the absence of abundant labeled data. When fine-tuned with supervised learning, the DBN becomes capable of leveraging both its generative foundations and discriminative power, making it especially effective for anomaly detection tasks in mobile communication networks where data is often noisy, high-dimensional, and only partially labeled [52–54].

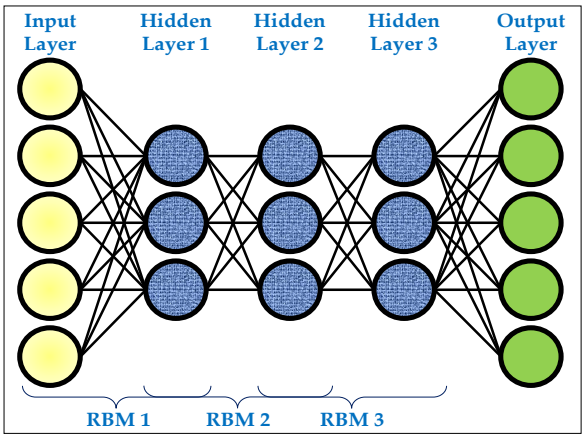

Figure 3. The generic structure of a DBN.

The foundation of a DBN lies in the RBM, which defines the relationship between visible and hidden variables. Equation (10) presents the energy function of an RBM, which assigns a scalar energy to each joint configuration of visible and hidden units:

$$E(v,h) = -v^T W h - b^T v - c^T h \tag{10}$$

Here, v denotes the visible vector; h the hidden vector; W the weight matrix connecting visible and hidden units; b and c represent the bias vectors for visible and hidden layers, respectively. Building upon the energy function, the joint probability distribution of

visible and hidden units is defined by the Boltzmann distribution. Equations (11) and (12) formalizes this distribution [53]:

$$P(v,h) = \frac{1}{Z} exp(-E(v,h)) \tag{11}$$

$$Z = \sum_{v,h} exp(-E(v,h)) \tag{12}$$

In this expression, Z is the partition function, which ensures that probabilities are properly normalized. From the joint distribution, the marginal probability of a visible vector can be derived. Equation (13) shows this marginal likelihood:

$$P(v) = \frac{1}{Z} \sum_{h} exp(-E(v,h)) \tag{13}$$

This expression quantifies how likely an observed input v is under the RBM model by summing over all possible hidden configurations. To enable efficient training, DBNs exploit the conditional independence between visible and hidden units. Equation (14) expresses the conditional probability of a hidden unit given the visible vector [54]:

$$P(h_j = 1 \mid v) = \sigma\left(\sum_i v_i W_{ij} + c_j\right) \tag{14}$$

where $\sigma(.)$ is the logistic sigmoid function. Similarly, the reconstruction of visible units from hidden states is captured in Equation (15):

$$P(v_i = 1 \mid h) = \sigma\left(\sum_j h_j W_{ij} + b_i\right) \tag{15}$$

These conditional probabilities provide the basis for the contrastive divergence learning algorithm, which approximates the gradient of the log-likelihood. Finally, by stacking multiple RBMs, a DBN is formed where the hidden activations of one layer serve as the visible inputs of the next. Equation (16) summarizes the layer-wise generative structure:

$$P(v,\ h^1, h^2,\ \ldots, h^L\ ) = P(v|\ h^1\ ) \prod_{l=1}^{L-1} P(h^l|\ h^{l+1}\ ) P(\ h^L\ ) \tag{16}$$

In this formulation, $h^1, h^2,\ \ldots, h^L$ represent the hidden layers of the DBN, with higher layers modeling increasingly abstract features [52].

2.4. IOA

The OA is a relatively recent population-based meta-heuristic optimization method, first introduced by Kaveh et al. in 2023 [55]. Inspired by the natural process of cultivating and managing orchards, it treats candidate solutions as trees that undergo different stages of growth and renewal. OA was designed to address complex optimization problems across engineering, computer science, and data-driven applications. Its main advantages include a balance between exploration and exploitation, the ability to escape local optima, and adaptability to both continuous and discrete search spaces. By mimicking realistic agricultural operations, OA offers a novel search dynamics compared to classical algorithms such as particle swarm optimization (PSO) or genetic algorithm (GA), leading to more diverse candidate solutions and improved convergence speed [55].

The algorithm proceeds through several operators in each iteration. The growth operator enables each solution to evolve toward stronger candidates by introducing controlled random variations. The screening operator evaluates and filters weaker trees to maintain population quality. The grafting operator combines genetic material from strong

and medium-quality candidates to generate promising offspring. The pruning operator randomly replaces underperforming solutions with new candidates to increase diversity. Finally, the elitism mechanism ensures that the best-performing solutions are preserved for the next generation. A single iteration therefore involves generating new solutions through growth and grafting, maintaining diversity via pruning, filtering candidates through screening, and securing the global best through elitism. This cycle is repeated until convergence, effectively imitating the management of a thriving orchard. The mathematical formulation of the OA begins with the growth operator. Equation (17) models the annual growth of a tree, where each candidate solution is updated by adding a growth factor and a random component that introduces variability in the growth direction. This mechanism ensures that solutions are not static but continuously adapt toward better regions of the search space [55].

$$X_{growth} = X_{current} + \delta R, \tag{17}$$

where X_{growth} is the new solution; $X_{current}$ is the current position; δ is the growth factor; R is a random variable introducing variability in the direction of growth.

The overall evaluation of solutions is governed by the fitness function. Equations (18)–(21) define the total fitness of each candidate by combining two complementary measures: the normalized fitness value and the growth rate over multiple years. Unlike many metaheuristics that rely solely on fitness magnitude, OA introduces a distinctive objective that simultaneously considers both the quality of the current solution and its growth trajectory across generations. This dual emphasis enables a more balanced exploration of the search space and provides resilience against premature convergence [55].

$$F_j = \alpha f'_j + \beta g'_j, \tag{18}$$

$$f'_j = \frac{f_j}{\max\limits_{1 \le j \le n} f_j}, \tag{19}$$

$$g_j = \sum_{i=m-l}^{m} \lambda_i \left(f_j^i - f_j^{i-1} \right), \tag{20}$$

$$g'_j = \frac{g_j}{\max\limits_{1 \le j \le n} g_j}, \tag{21}$$

where F_j is the total fitness function; f_j is the fitness function value of j-th candidate; f'_j is the normalized f_j; g_j is the growth rate of solution j; g'_j is the normalized g_j; m is the total number of growth years; α and β are weighting factors; λ is the weight given to those years; l is the number of growth years.

Beyond growth and fitness evaluation, additional operators further refine the population. Equation (22) formulates the grafting process, which blends the characteristics of a strong candidate with those of a medium-quality one to generate new offspring. This controlled recombination promotes the inheritance of favorable traits while maintaining diversity. Equation (23) represents the pruning step, where weak or stagnant candidates are replaced by randomly generated solutions within the search boundaries, ensuring the population does not lose exploratory capability [55].

$$X_{grafted} = \theta . X_{strong} + (1 - \theta) . X_{medium}, \tag{22}$$

$$X_{pruned} = Random(X_{min}, X_{max}) \tag{23}$$

where $X_{grafted}$ is the grafting solution; X_{medium} is the position of the medium-quality candidate; X_{strong} is the position of the stronger candidate; θ is a blending coefficient

determining the contribution of each parent; X_{min} *and* X_{max} are the bounds of the search space; $X_{pruning}$ is the new randomly generated solution.

Through these mathematical formulations, the OA completes a cycle of growth, evaluation, grafting, and pruning, while elitism preserves the best-performing solutions for the next iteration. This structured process provides an effective balance between intensification and diversification, driving the search toward global optima. Although the OA offers a fresh perspective on population-based optimization, it still faces challenges in fully balancing exploration and exploitation. In some cases, the growth and grafting processes may bias the search toward intensification around promising solutions, reducing the diversity needed to escape local optima. Conversely, the pruning and random replacement steps, while useful for exploration, can occasionally disrupt convergence by introducing excessive randomness. As a newly developed algorithm, its mechanisms for maintaining this delicate balance are not yet as mature or well-validated as in more established meta-heuristics, which means there is still significant potential for refinement and extension to improve its robustness across complex optimization landscapes.

One of the main challenges of the OA lies in its limited ability to consistently balance exploration and exploitation. While growth and grafting drive intensification around promising solutions, pruning and random replacement sometimes create excessive randomness, weakening convergence. To address this limitation, a novel extension of the OA is introduced by incorporating a seasonal migration operator inspired by the natural cycles of seasonal change in orchards. This operator is specifically designed to reinforce both exploration and exploitation in a controlled and adaptive manner, thereby overcoming the tendency of the original algorithm to either converge prematurely or drift randomly in the search space.

The seasonal migration operator functions by alternately guiding candidate solutions toward either the global best solution (exploitation) or distant, less similar solutions (exploration). By probabilistically switching between these two modes, the operator maintains population diversity while simultaneously intensifying the search around high-quality regions. When the probability favors exploitation, the search is refined around the global best, accelerating convergence. Conversely, when the probability favors exploration, the operator directs the search toward distant candidates, effectively expanding the search horizon and reducing the risk of stagnation. This adaptive dual behavior ensures that both key components of meta-heuristic optimization are continuously addressed throughout the iterative process. Formally, the seasonal migration operator is defined in Equations (24) and (25):

$$X_i^{t+1} = \begin{cases} X_i^t + \eta\left(X_{best}^t - X_i^t\right) & with\ probability\ p_t, \\ X_i^t + \eta\left(X_{far}^t - X_i^t\right) & with\ probability\ 1 - p_t \end{cases} \tag{24}$$

$$p_t = p_{min} + (p_{max} - p_{min})\left(\frac{t}{T}\right)^{\tau} \tag{25}$$

where X_i^t is the position of solution i at iteration t; X_{best}^t denotes the current global best solution; X_{far}^t is a candidate selected from the most dissimilar individuals in the population; the parameter η is a learning rate controlling the migration step size; p_t is the probability of exploitation versus exploration; T is the total number of iterations; $p_{min} = 0.20$; $p_{min} = 0.85$; $\tau = 1.5$ controls the curvature of the transition. Algorithm 1 presents pseudo-code of IOA.

Algorithm 1 Pseudo-codes of improved OA

Begin IOA
%%Setting
Initialization of number of strong/medium/weak trees, α, β, population size, Iteration
%%Initial population
for n = 1 to population size do
Create population (trees)
Calculate the fitness function (f_j)
end
%%Loop
for i = 1 to Max iteration do
%% Elitism
Sort population
Select elites
%% Growth
for j = 1 to population size do
$X_{growth} = X_{current} + \delta R$
Calculate the fitness function (f_j)
end
%% Screening
Save previous populations for growth rate
for j = 1 to population size do
Calculate the normalized fitness function (f'_j) based Equation (19)
Calculate the normalized growth rate (g'_j) based Equation (21)
Calculate the total fitness function (F_j) based Equation (18)
Screening: strong, medium, and weak
end
%% Grafting
for j = 1 to population size do
$X_{grafted} = \theta.X_{strong} + (1-\theta).X_{medium}$
Calculate the fitness function (f_j)
end
%% Pruning
for p = 1 to population size do
$X_{pruned} = Random(X_{min},\ X_{max})$
Calculate the fitness function (f_j)
end
%% Seasonal Migration
for j = 1 to population size do

$$X_i^{t+1} = \begin{cases} X_i^t + \eta\left(X_{best}^t - X_i^t\right) & with\ probability\ p, \\ X_i^t + \eta\left(X_{far}^t - X_i^t\right) & with\ probability\ 1-p \end{cases}$$

Calculate the fitness function (f_j)
end
Sort the new population
Show the best solution
end
End IOA

By incorporating this operator into OA, the improved framework adaptively alternates between convergence and diversification, leading to a more reliable and effective optimiza-

tion process. From an optimization-theoretic standpoint, the seasonal-migration operator serves as a controlled symmetry-breaking mechanism within the evolutionary population. During early iterations, when the candidate solutions tend to cluster around symmetric basins of attraction (regions representing repetitive or equivalent states in the search space) the exploratory branch introduces deliberate asymmetry by guiding individuals toward distant, non-similar solutions. This mechanism disrupts excessive regularity and injects diversity, preventing the optimizer from being confined to repetitive trajectories or shallow local minima. In later iterations, when convergence becomes desirable, the exploitative branch restores useful symmetries by aligning the population around the best-performing configurations. Hence, the operator dynamically alternates between symmetry preservation and symmetry breaking, enabling the IOA to maintain a self-regulated balance between exploration and exploitation.

This adaptive interplay between symmetric and asymmetric behaviors has broader implications beyond the current anomaly detection task. In complex engineered systems (such as EDA processes) optimization problems often involve symmetric structures (e.g., mirrored circuit topologies, balanced transistor pairs, or repeated layout cells) that ensure predictable and stable performance. However, controlled asymmetry is equally valuable, as it allows the design process to escape performance plateaus or to compensate for process variations and parasitic mismatches. The same principle realized in the IOA's seasonal-migration operator can thus be leveraged to guide symmetry-aware optimization in circuit or system design, where breaking non-beneficial regularities while preserving functional balance leads to superior efficiency, robustness, and adaptability.

2.5. Proposed HTST-DBN

Figure 4 depicts the end-to-end HTST-DBN pipeline. The TST encoder converts preprocessed sequences into context-aware embeddings via multi-head self-attention (Q/K/V projections, scaled dot-product attention, feed-forward blocks, and residual normalization). The IOA operates outside the forward path to tune hyper-parameters and architectural knobs. The DBN consumes the TST embeddings and learns hierarchical, generative abstractions through stacked RBMs. The right-hand panel highlights the Q/K/V flow culminating in an anomaly score (or class) at the output. In the proposed HTST-DBN, the TST is responsible for modeling temporal dependencies and contextual interactions in network traffic/telemetry. It projects each sequence into a fixed-dimensional space, injects temporal order via positional encodings, and aggregates information across time using multi-head attention, enabling the model to capture both short-term bursts (e.g., DDoS spikes) and long-range trends (e.g., slow drift in IoT sensors). This is essential for heterogeneous, noisy streams typical of mobile networks, where dependencies span multiple time scales and sources.

The DBN serves as a deep feature abstraction module on top of the TST embeddings. By stacking RBMs and leveraging generative pretraining followed by fine-tuning, the DBN uncovers latent structure and denoises the representation, yielding features that are robust to sparsity, missing values, and class imbalance. In practice, the DBN sharpens the separation between normal and anomalous regimes. IOA performs hyper-parameter/architecture search for the hybrid stack, optimizing (for example) TST depth/width, attention heads, learning rates, DBN layer sizes, and regularization strength. Unlike conventional optimizers, IOA evaluates candidates using a composite objective that accounts not only for the current validation fitness but also for the growth rate of that fitness across recent "years" (iterations), as formalized in Equations (18)–(21). To further strengthen search dynamics, a Seasonal Migration operator is incorporated, probabilistically alternating movements toward the global best (exploitation) and toward distant, dissimilar candidates (exploration).

Together with growth, grafting, pruning, screening, and elitism, this yields a stable yet diverse search over the TST–DBN design space.

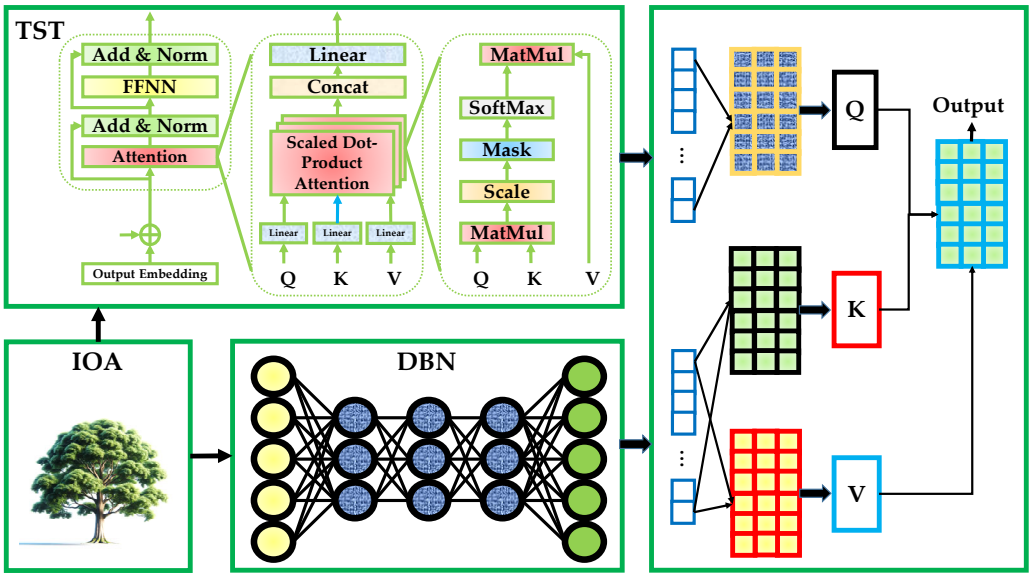

Figure 4. The proposed HTST-DBN architecture.

The hybrid design exploits complementary strengths: TST excels at sequence-level context aggregation; DBN excels at generative abstraction and denoising; IOA adaptively discovers configurations that generalize across heterogeneous data. Empirically, this synergy improves detection accuracy, stability under noise/imbalance, and transferability across datasets, yielding a practical, scalable solution for anomaly detection in next-generation mobile networks.

To integrate the temporal representation of the TST with the hierarchical abstraction capability of the DBN, an intermediate projection mechanism was applied. The final encoder output of TST, denoted as $\in \mathbb{R}^{L \times d_{model}}$, was first flattened into a one-dimensional vector through a global average pooling operation across the temporal axis, resulting in a feature vector $Z' \in \mathbb{R}^{d_{model}}$. When the temporal length L exceeded 50, a dense projection layer with a learnable weight matrix $W_P \in \mathbb{R}^{d_{model} \times d_{DBN}}$ was employed to map the transformer embedding space into the input dimension of the first RBM in the DBN. In our implementation, $d_{model} = 128$ and $d_{DBN} = 100$, ensuring smooth dimensional continuity between the two submodules. This projection not only compresses the temporal embeddings into a compact feature representation but also preserves high-order dependencies learned by the TST attention heads before hierarchical feature abstraction in the DBN model.

To maintain gradient stability during joint training and to prevent vanishing gradients across the deep hierarchical layers, several optimization strategies were integrated into the HTST-DBN training process. First, residual normalization was preserved in each TST encoder block, which allows stable gradient propagation through the self-attention layers. Second, the DBN parameters were pre-trained layer by layer using the Contrastive Divergence (CD-1) algorithm before end-to-end fine-tuning, ensuring that the initial weights of each RBM layer were near an optimal energy state. During joint optimization, dropout (rate = 0.3) and adaptive learning-rate scheduling (initialized at 10^{-3}) were employed to mitigate gradient explosion and ensure smooth convergence. Additionally, the IOA dynamically tuned the learning rates and weight decay parameters of both sub-networks, allowing synchronized updates between the Transformer and DBN during fine-tuning. These mechanisms collectively stabilized the backward pass, accelerated convergence, and effectively eliminated gradient vanishing even under deep model configurations.

3. Results

All experiments were implemented in Python 3.10, using a combination of well-established libraries tailored to deep learning and optimization. The hybrid framework was built with TensorFlow 2.11 and Keras 2.11 for implementing the Time Series Transformer and DBN components, while NumPy 1.23, Pandas 1.5, and Scikit-learn 1.2 were employed for data preprocessing, normalization, and evaluation. For optimization and metaheuristic routines, including the IOA, custom modules were developed and integrated with SciPy 1.10. Visualization and performance analysis were carried out using Matplotlib 3.7 and Seaborn 0.12, ensuring reproducibility of figures and statistical plots. The experiments were executed on a workstation equipped with an Intel Core i9-12900K CPU @ 3.2 GHz, 64 GB RAM, and an NVIDIA RTX 3090 GPU with 24 GB memory. This hardware configuration provided sufficient computational capacity to train the hybrid HTST-DBN model on large-scale datasets such as CICDDoS2019 and ToN-IoT, while also supporting extensive hyper-parameter tuning through IOA.

To comprehensively evaluate the effectiveness of the proposed HTST-DBN framework, its performance was benchmarked against a diverse set of state-of-the-art algorithms, including TST, BERT, DBN, DRL, CNN, and RF. These baselines were chosen to represent different families of anomaly detection approaches: transformer-based architectures (TST, BERT), deep generative networks (DBN), reinforcement learning (DRL), convolutional networks (CNN), and classical ML classifiers (RF). Comparing against this broad spectrum allows us to validate the generalization ability and robustness of the proposed hybrid design. In particular, TST and DBN were also evaluated as standalone models in order to assess their individual strengths and weaknesses, and to demonstrate the incremental benefits of combining temporal modeling with deep feature abstraction.

Each baseline contributes unique capabilities that make the comparison both fair and insightful. TST captures long-range temporal dependencies in sequential data, while BERT offers powerful contextual representation learning adapted from NLP to sequential network traffic. DBN is effective in generative feature learning, while DRL provides adaptive decision-making under dynamic environments, reflecting realistic mobile and IoT network scenarios. CNN excels at extracting spatially local patterns and has proven effective in intrusion detection, while RF serves as a strong classical benchmark due to its ability to handle noisy and heterogeneous features. Beyond these baselines, the hybrid model was also trained using alternative meta-heuristic algorithms for hyper-parameter optimization, such as GA and PSO, to explicitly demonstrate the advantages of the proposed IOA. This comparative design ensures that improvements observed with HTST-DBN are not simply due to architectural complexity but stem from the synergy between TST, DBN, and IOA, making the evaluation rigorous and comprehensive.

For the evaluation of all algorithms, a comprehensive set of performance metrics was employed, including accuracy, recall, area under the curve (AUC), root mean square error (RMSE), execution time, variance, and the t-test. Together, these metrics allow us to assess not only classification effectiveness but also the stability, computational efficiency, and statistical reliability of the proposed HTST-DBN compared to the baselines. Equation (26) defines accuracy, which measures the overall proportion of correctly classified instances, combining both true positives and true negatives over the total population. A higher accuracy indicates that the model achieves a strong balance in correctly detecting normal and anomalous classes.

$$Accuracy = \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negative} \tag{26}$$

Equation (27) describes recall, which measures the ability of the model to identify actual anomalies from all anomalous samples present in the dataset. High recall indicates robustness against missed detections, which is particularly important in security applications.

$$Recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{27}$$

Equation (28) introduces AUC, which corresponds to the area under the receiver operating characteristic (ROC) curve and provides a threshold-independent evaluation of classification performance. A higher AUC value reflects better discrimination between normal and attack classes across all thresholds.

$$AUC = \int_0^1 ROC(t)dt \tag{28}$$

where $ROC(t)$ is the ROC curve at threshold t.

Equation (29) defines RMSE, which is employed to track the training process by quantifying the average squared deviation between predicted and observed labels. Lower RMSE values indicate faster convergence and more stable learning, making it an important measure for evaluating optimization quality in the training phase.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{y}_i]^2}, \tag{29}$$

where y_i is the observed value; $\hat{y}_i$ is the calculated value.

Beyond these formula-based metrics, execution time is measured to assess the computational cost of each algorithm, reflecting efficiency in large-scale deployment. Variance captures the stability of results across multiple runs, where lower variance denotes consistency and reliability in the model's predictions. Finally, the t-test is conducted as a statistical significance test to verify whether the improvements achieved by the model over baselines are not due to random chance. Before presenting the quantitative results, it is essential to emphasize the role of parameter tuning, since the effectiveness of deep models strongly depends on properly chosen hyper-parameters. The HTST-DBN benefits from the IOA, which adaptively optimizes parameters such as learning rate and attention heads. In contrast, the baseline models including TST, BERT, DBN, DRL, CNN, and RF were tuned using the conventional grid search method. This distinction highlights the advantage of IOA in dynamically balancing exploration and exploitation, yielding parameter settings that are more tailored to the data and the optimization landscape.

As shown in Table 2, the configurations of all models are reported separately to ensure a fair and reproducible comparison. For the proposed HTST-DBN, the hyper-parameter calibration was conducted through the IOA, which adaptively searched within predefined ranges for each parameter. The learning rate was explored in the interval $[10^{-5}, 10^{-2}]$ and optimized to 0.004, while the batch size was tuned within $\{16, 32.64, 128\}$, converging to 32 as the most stable setting. The TST component was configured with 8 encoder layers and 8 attention heads, selected from search bounds of 4–10 layers and 4–16 heads, respectively. The DBN component comprised 4 hidden layers, each containing 32 neurons, optimized within 2–5 layers and 16–64 neurons per layer, using Tanh and Sigmoid activations to balance non-linear abstraction and stability. Dropout was maintained at 0.2, selected from a search range of 0.1–0.4 to minimize overfitting without compromising generalization. For the IOA parameters, the population size was optimized to 110 (search range 80–140) and the maximum number of iterations fixed at 300 (range 200–400). The weighting coefficients were dynamically adjusted within the ranges $\alpha \in [0.5,\ 0.8]$ and $\beta \in [0.2,\ 0.4]$, settling at

$\alpha = 0.68$ and $\beta = 0.32$, which yielded the most balanced trade-off between exploration and exploitation.

Table 2. Hyper-parameter settings of the proposed HTST-DBN and baseline models.

Model	Parameter	Value
HTST-DBN	Learning rate	0.004
	Batch size	32
	Number of attention heads	8
	Number of transformer layers	8
	Dropout rate	0.2
	Number of hidden layers	4
	Number of neurons in hidden layers	32
	Activation	Tanh and sigmoid
	Optimizer	IOA
	Number of strong trees	40
	Number of medium trees	42
	Number of weak trees	28
	α	0.68
	β	0.32
	Population size	110
	Iteration	300
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
DRL	Learning rate	0.002
	Discount factor (γ)	0.95
	ε-greedy	0.44
	Batch size	64
CNN	Activation Function	Sigmoid
	Number of convolution layers	8
	Kernel size	5×5
	Pooling type	Max pooling (2×2)
	Number of hidden layers	6
	Number of neurons	64
	Activation	Tanh
	Optimizer	Adam
RF	Criterion	Gini
	Number of estimators	300
	Minimum samples per split	6
	Maximum depth of trees	10

BERT used a learning rate of 0.002, batch size 32, dropout 0.1, 6 self-attention heads, and 6 transformer encoder layers, with an input sequence window of 64 and the GELU activation function trained with SGD. DRL was configured with a learning rate of 0.002, discount factor of 0.95, ϵ-greedy value of 0.44, and batch size 64. CNN employed 8 convolution layers with kernel size 5×5, max pooling of 2×2, and 6 hidden layers with 64 neurons each. Finally, RF was set with 300 estimators, a minimum of 6 samples per split, and a maximum depth of 10, using Gini as the impurity criterion. These parameter values underline the structural and algorithmic diversity across the baselines, ensuring that the comparative results are based on competitive configurations for each model.

Table 3 shows the comparative performance of the proposed HTST-DBN model and the baseline algorithms across the four benchmark datasets. As shown in Table 3, the performance comparison highlights the superiority of the proposed HTST-DBN framework

over all baseline methods across four benchmark datasets. Each dataset is evaluated using Accuracy, Recall, and AUC, providing a multi-perspective assessment of detection capability. On the CICDDoS2019 dataset, HTST-DBN achieves 99.61% accuracy, 99.85% recall, and 99.93% AUC, outperforming the strongest baseline (TST with 90.28% accuracy and 92.20% AUC) by nearly 9–10 percentage points. This large margin is explained by the hybrid design, where TST captures bursty sequential patterns of DDoS flows while DBN enhances robustness by abstracting noisy features. In contrast, CNN and RF lag significantly with accuracies of 84.86% and 81.08%, highlighting the limitations of shallow or static architectures when facing high-volume attack traffic.

Table 3. Comparative performance of the proposed HTST-DBN and baseline models.

Model	Datasets											
	CICDDoS2019			Edge-IoTset			5G-NIDD			ToN-IoT		
	Accuracy	Recall	AUC	Accuracy	Recall	AUC	Accuracy	Recall	AUC	Accuracy	Recall	AUC
HTST-DBN	99.61	99.85	99.93	99.58	99.76	99.82	99.52	99.73	99.78	99.48	99.60	99.86
TST	90.28	91.25	92.20	89.70	90.23	91.43	88.22	89.18	90.14	87.42	88.20	89.33
BERT	89.40	90.35	91.86	88.25	89.53	90.33	89.29	90.24	91.26	86.18	87.47	88.25
DBN	88.53	89.41	90.51	87.37	88.45	89.62	86.85	87.09	88.22	85.63	86.29	87.76
DRL	87.18	88.25	89.09	86.55	87.76	88.18	87.45	88.45	89.54	84.43	83.05	84.18
CNN	84.86	85.36	86.48	83.08	84.32	85.72	82.30	83.32	84.71	80.83	81.13	82.37
RF	81.08	82.90	83.27	80.76	81.05	82.19	79.77	80.10	81.88	78.20	79.58	80.22

For Edge-IoTset and 5G-NIDD, which are more complex due to heterogeneous IoT telemetry and 5G control-plane anomalies, HTST-DBN maintains outstanding performance with accuracies of 99.58% and 99.52%, recalls above 99.7%, and AUC values close to 99.8%. Baselines such as BERT and TST show relatively strong results (around 88–90% accuracy), yet they fall short of the hybrid approach. The consistent improvement demonstrates that the integration of DBN with TST embeddings enables the model to generalize effectively in dynamic, edge-enabled and cellular environments. IOA's optimization further ensures that architecture depth, attention heads, and learning parameters are well-tuned to maximize detection capability across diverse data modalities. The ToN-IoT dataset, being the most challenging due to its multi-source, noisy, and imbalanced nature, still shows the dominance of HTST-DBN with 99.48% accuracy, 99.60% recall, and 99.86% AUC. In contrast, the best-performing baseline (TST) achieves only 87.42% accuracy and 89.33% AUC, while RF and CNN remain below 81% accuracy. This highlights how the hybrid model mitigates the difficulties of multi-source telemetry by leveraging both temporal dependencies and probabilistic abstractions.

Figure 5 shows the bar chart representation of the comparative results previously reported in Table 3. Specifically, it illustrates how the proposed HTST-DBN consistently achieves the highest Accuracy, Recall, and AUC across all four datasets, compared with baseline models including TST, BERT, DBN, DRL, CNN, and RF. By transforming the tabular results into a visual format, the superiority of the hybrid approach becomes more evident, highlighting the significant margins of improvement in both detection effectiveness and robustness across diverse environments. The visualized results emphasize how the integration of TST, DBN, and IOA enables the HTST-DBN to capture long-range temporal dependencies, extract robust latent features, and fine-tune hyper-parameters in a way that directly addresses the challenges of anomaly detection in mobile and IoT networks. TST alone captures sequential dependencies but misses deeper abstraction; DBN alone captures latent features but lacks temporal sensitivity; and IOA ensures optimal balance of depth, attention, and regularization. Their combination leads to a powerful synergy where anomalies are detected with near-perfect reliability, even in the noisy and heterogeneous

ToN-IoT dataset. This validates the novelty of the proposed model in effectively solving the problem of anomaly detection across multiple real-world scenarios.

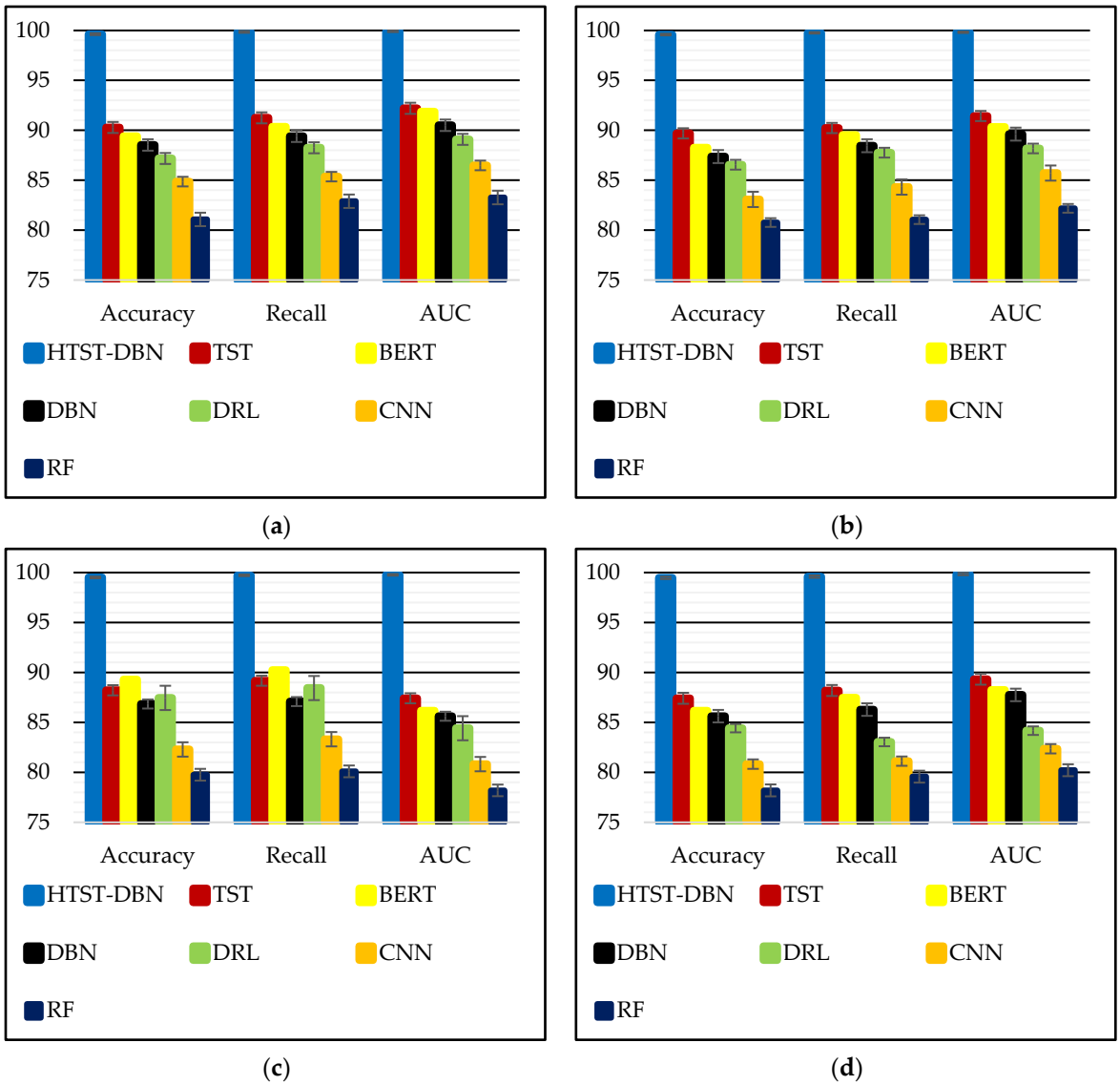

Figure 5. Bar chart visualization of accuracy, recall, and AUC across four datasets: (**a**) CICDDoS2019; (**b**) Edge-IoTset; (**c**) 5G-NIDD; (**d**) ToN-IoT.

Figure 6 shows the ROC curves of the proposed HTST-DBN and baseline models, illustrating their ability to balance sensitivity (true positive rate) against 1-specificity (false positive rate) across varying thresholds. The steeper and more convex curve of HTST-DBN in all four subplots indicates its superior discriminative capability, consistently approaching the top-left corner of the plot. This demonstrates that HTST-DBN maintains a high detection rate while minimizing false alarms, a critical requirement in anomaly detection systems for communication networks.

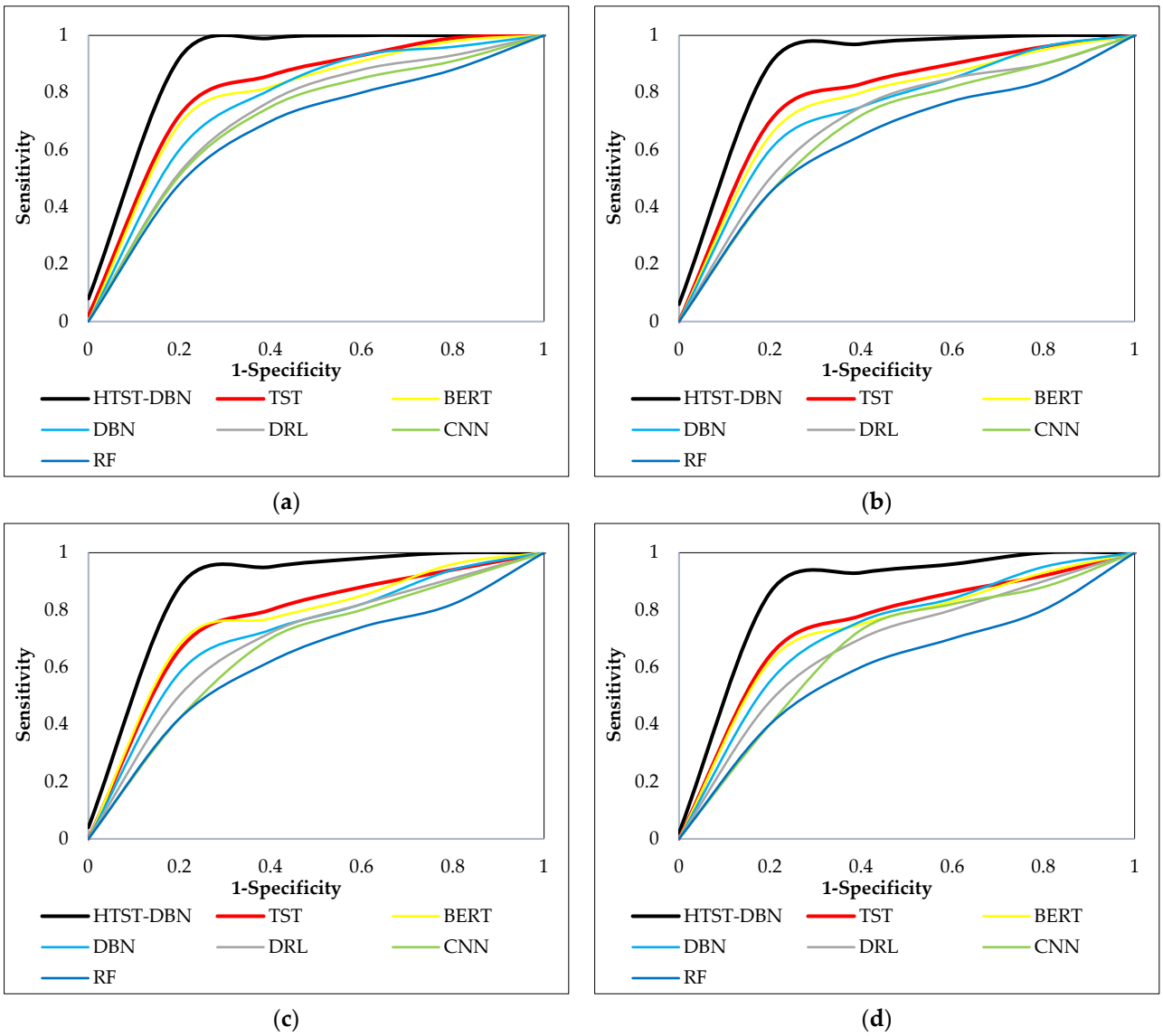

Figure 6. ROC curve comparisons across four benchmark datasets: (**a**) CICDDoS2019; (**b**) Edge-IoTset; (**c**) 5G-NIDD; (**d**) ToN-IoT.

From a technical perspective, the curvature of HTST-DBN's ROC reflects the synergy between temporal encoding, probabilistic abstraction, and IOA-driven hyper-parameter optimization, which together produce a more robust decision boundary. In contrast, baselines such as RF and CNN exhibit flatter curves, reflecting weaker sensitivity–specificity trade-offs and higher susceptibility to misclassification under threshold variation. Even strong models like TST and BERT show less convex ROC profiles compared to HTST-DBN, underlining the latter's advantage in threshold-independent evaluation. These results confirm that the proposed architecture not only improves point metrics like accuracy and recall but also achieves consistently superior classification robustness across a full spectrum of operating conditions.

Table 4 presents the comparative analysis of the proposed HTST-DBN framework against multiple architectural and optimization configurations, combining TST and DBN components with different meta-heuristic algorithms, including IOA, OA, PSO, GA, and chimp optimization algorithm (ChOA). This design enables a systematic evaluation of both architectural contributions and optimizer effects. By contrasting the IOA-based model with

traditional optimization techniques, the table provides a holistic perspective on how search strategies influence convergence quality and classification performance across datasets.

Table 4. Comparative analysis of different architectural combinations and optimization strategies.

Model	Datasets											
	CICDDoS2019			Edge-IoTset			5G-NIDD			ToN-IoT		
	Accuracy	Recall	AUC	Accuracy	Recall	AUC	Accuracy	Recall	AUC	Accuracy	Recall	AUC
HTST-DBN (IOA)	99.61	99.85	99.93	99.58	99.76	99.82	99.52	99.73	99.78	99.48	99.60	99.86
HTST-DBN (OA)	96.75	97.15	97.97	96.08	96.74	97.14	95.31	96.18	97.05	94.35	95.73	96.67
HTST-DBN (PSO)	96.08	96.82	97.92	95.14	96.15	97.82	94.17	95.26	96.40	93.12	94.14	95.28
HTST-DBN (ChOA)	95.86	96.43	97.88	94.88	95.93	96.72	94.09	95.19	95.98	92.86	93.48	95.34
HTST-DBN (GA)	95.36	96.35	97.80	94.50	95.72	96.46	93.91	94.60	95.44	92.27	93.38	94.41
IOA-TST	94.46	95.28	96.16	93.46	94.29	95.50	92.38	93.76	94.52	91.33	92.42	93.78
IOA-DBN	93.20	94.18	95.71	92.34	93.05	94.34	91.21	92.37	93.27	90.04	91.26	92.92
TST-DBN	92.19	93.21	94.24	91.11	92.22	93.80	90.05	91.18	92.02	89.15	90.44	91.35
TST	90.28	91.25	92.20	89.70	90.23	91.43	88.22	89.18	90.14	87.42	88.20	89.33
DBN	88.53	89.41	90.51	87.37	88.45	89.62	86.85	87.09	88.22	85.63	86.29	87.76

Quantitatively, IOA delivers the most consistent and superior results, with accuracies exceeding 99.4% and AUC values approaching 99.9% on all datasets. For instance, on CICDDoS2019, IOA achieves 99.61% accuracy, 99.85% recall, and 99.93% AUC, outperforming PSO (96.08% accuracy), GA (95.36%), OA (96.75%), and ChOA (95.86%). This clearly demonstrates IOA's enhanced exploration–exploitation balance, achieved through its adaptive migration probability and seasonal learning operator, which accelerate convergence and prevent premature stagnation. In contrast, standard OA lacks dynamic parameter adaptation, leading to slower convergence, while ChOA improves exploration via multi-group communication but still exhibits weaker exploitation consistency than IOA. Classical PSO and GA show moderate convergence speeds but struggle in maintaining solution diversity, which reduces their ability to escape local minima.

Furthermore, hybrid architectural variants such as TST-DBN and IOA-DBN perform better than single-component models but remain inferior to the full HTST-DBN configuration. For instance, TST-DBN attains 92.19% accuracy on CICDDoS2019 and 89.15% on ToN-IoT, while IOA-DBN slightly improves to 93.20% and 90.04%, respectively. These findings confirm that both the architectural integration and the optimization mechanism are essential. The combination of TST's temporal feature extraction, DBN's deep abstraction, and IOA's adaptive optimization enables the proposed HTST-DBN to achieve robust and scalable anomaly detection performance across diverse network scenarios.

Figure 7 shows the evolution of RMSE with respect to training epochs, illustrating how quickly and effectively each model converges. The proposed HTST-DBN consistently achieves the fastest convergence, with RMSE dropping sharply within the first 40–60 epochs across all datasets and reaching near-zero values well before 100 epochs. In contrast, other deep models such as TST, BERT, and DBN require over 150 epochs to stabilize at higher RMSE values, while CNN and RF converge much more slowly and remain at significantly higher error levels. This demonstrates the efficiency of the hybrid framework in learning robust patterns without prolonged training cycles. An additional observation is that the IOA-driven optimization not only accelerates convergence but also reduces oscillations in the learning curves. While models like DRL and CNN exhibit fluctuating RMSE values during early epochs, HTST-DBN maintains a smooth downward trajectory, reflecting more stable gradient updates and better control over overfitting. This stability is particularly important in large-scale datasets such as CICDDoS2019 and ToN-IoT, where noisy and imbalanced data can otherwise cause unstable training behavior.

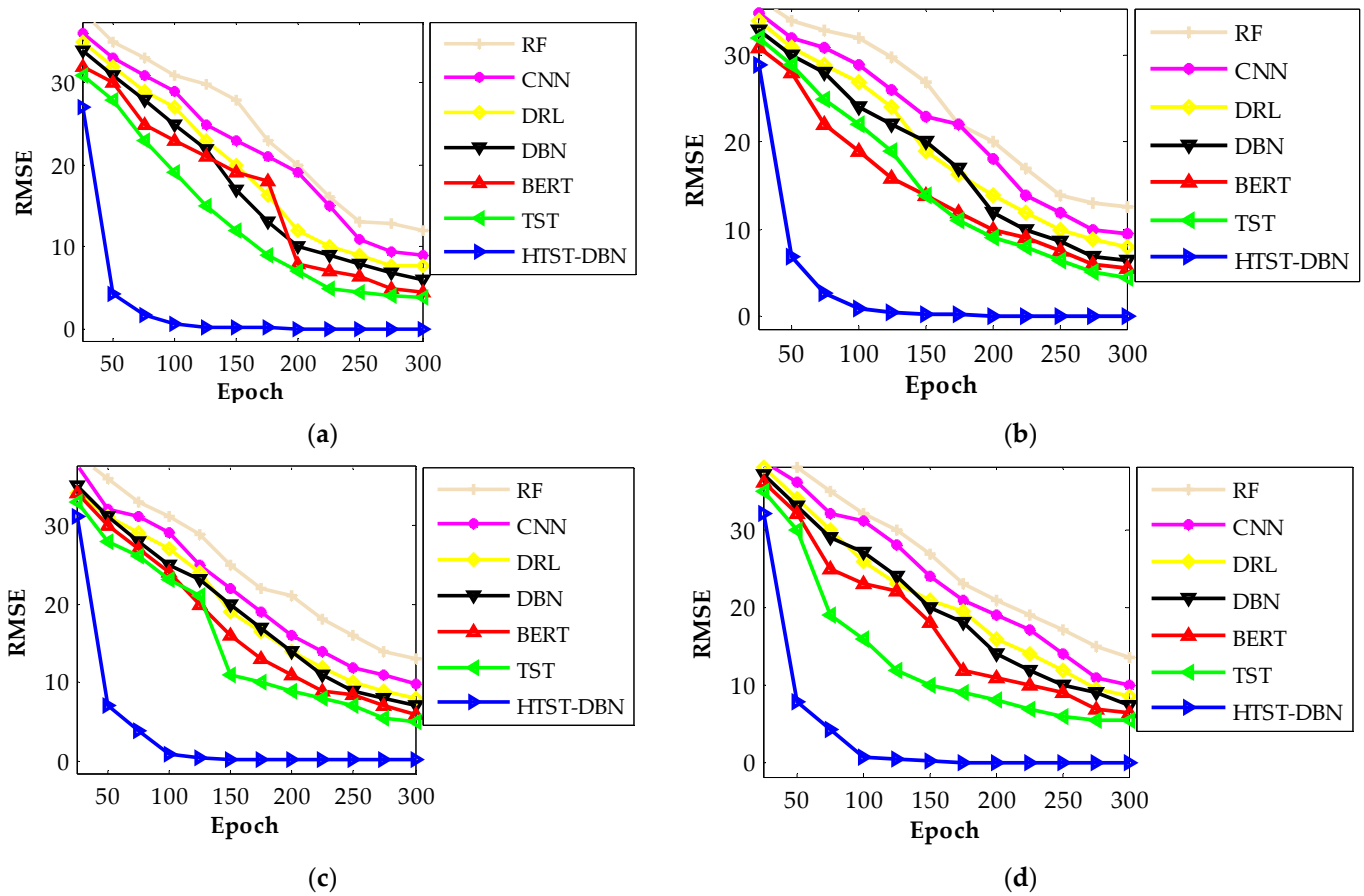

Figure 7. Training convergence curves of RMSE across four datasets: (**a**) CICDDoS2019; (**b**) Edge-IoTset; (**c**) 5G-NIDD; (**d**) ToN-IoT.

A closer analysis across datasets reveals consistent trends: on CICDDoS2019 and Edge-IoTset, HTST-DBN achieves full convergence by approximately epoch 70, whereas baselines like TST and DBN flatten around epoch 200. For 5G-NIDD and ToN-IoT, which are inherently more complex and noisy, HTST-DBN still converges by epoch 90, while the baselines require nearly the entire 300-epoch span to stabilize. This accelerated convergence highlights the effectiveness of IOA-driven parameter tuning combined with the complementary strengths of TST and DBN, enabling the model not only to learn faster but also to generalize more effectively with minimal training error.

Table 5 presents a comprehensive comparison between the proposed HTST-DBN framework and several advanced deep-learning architectures inspired by recent studies discussed in the related-work section. To ensure fairness and consistency, these comparative models (including ABCNN [30], DRL-RBFNN [34], OS-CNN [41], DeepTransIDS [36], DNN [43], and RF-KNN [38]) were re-implemented and trained on the same benchmark datasets (CICDDoS2019, Edge-IoTset, 5G-NIDD, and ToN-IoT) under identical preprocessing and experimental conditions. Therefore, the reported metrics represent results obtained from our own experimental framework rather than the original studies, providing a direct and unbiased assessment of each model's generalization capability. Across all four datasets, HTST-DBN consistently achieves the best performance in terms of accuracy, recall, and AUC, confirming its superior detection capability and robustness. On the challenging CICDDoS2019 dataset, it reaches 99.61% accuracy and 99.93% AUC, surpassing the second-best ABCNN and DRL-RBFNN models by more than 6%. A similar trend is observed for Edge-IoTset, where HTST-DBN maintains 99.58% accuracy and 99.82% AUC, outperforming DeepTransIDS by roughly 9%, highlighting its resilience to heterogeneous

IoT telemetry. For 5G-NIDD, the proposed model attains 99.52% accuracy and 99.78% AUC, demonstrating effective adaptation to control-plane traffic dominated by short-sequence signaling. Even in the highly imbalanced ToN-IoT dataset, HTST-DBN achieves 99.48% accuracy and 99.86% AUC, clearly ahead of all baselines.

Table 5. Comparative performance of the proposed HTST-DBN and recent models implemented based on related works.

Model	Datasets											
	CICDDoS2019			Edge-IoTset			5G-NIDD			ToN-IoT		
	Accuracy	Recall	AUC	Accuracy	Recall	AUC	Accuracy	Recall	AUC	Accuracy	Recall	AUC
HTST-DBN	99.61	99.85	99.93	99.58	99.76	99.82	99.52	99.73	99.78	99.48	99.60	99.86
ABCNN [30]	93.32	94.05	94.86	92.19	93.54	94.28	91.17	92.08	93.17	90.25	91.54	92.31
DRL-RBFNN [34]	93.05	93.88	94.25	91.94	92.87	93.64	91.08	92.34	93.18	90.27	91.29	92.04
OS-CNN [41]	92.33	93.70	94.53	91.24	91.97	92.59	90.26	91.47	92.63	89.27	90.09	91.47
DeepTransIDS [36]	91.20	92.34	93.28	90.64	91.47	92.36	89.88	90.09	91.24	88.19	89.61	90.08
DNN [43]	89.32	90.69	91.75	88.09	89.34	90.60	87.09	88.62	89.19	86.51	87.09	88.37
RF-KNN [38]	88.24	89.35	90.04	87.67	88.41	89.37	86.24	88.19	88.37	85.19	86.92	87.16

4. Discussion

In the Results section, the superiority of the proposed HTST-DBN framework was demonstrated through the analysis of core evaluation metrics, including accuracy, recall, AUC, and training convergence behavior. The model consistently outperformed baseline approaches across all benchmark datasets, showing not only higher detection rates but also faster and more stable convergence. These findings underline the strength of combining temporal feature extraction through TST, deep generative abstraction with DBN, and IOA-driven hyper-parameter optimization. Together, these elements enabled the model to achieve robust classification performance across heterogeneous and noisy environments, validating its effectiveness as an anomaly detection solution for next-generation networks.

While these metrics highlight the predictive power of the proposed architecture, real-world applicability also depends on additional aspects such as variance across runs, statistical reliability through *t*-tests, and execution time efficiency. By examining variance, the stability and reproducibility of the model across multiple training sessions can be assessed, ensuring that the results are not overly sensitive to initialization. The *t*-test provides statistical confirmation that improvements over baselines are significant rather than random fluctuations. Moreover, execution time is a practical consideration for deployment in time-sensitive environments such as mobile edge or IoT networks. By analyzing these complementary dimensions, the evaluation determines whether HTST-DBN is not only accurate but also efficient, stable, and scalable for real-world applications.

Table 6 shows the robustness of the proposed HTST-DBN compared to baseline models, measured as the variance of results over 30 independent runs. From the results, HTST-DBN achieves remarkably low variance across all datasets, with values ranging from 0.00035 on CICDDoS2019 to 0.00072 on ToN-IoT, several orders of magnitude smaller than competing methods. In contrast, traditional baselines such as RF and CNN show significantly higher variance, exceeding 9.41 and 6.02, respectively, on CICDDoS2019, and surpassing 10.79 and 7.84 on ToN-IoT. Even stronger baselines like TST and BERT record variances above 2.1 and 2.8, underscoring that while they achieve reasonable accuracy, their training outcomes remain unstable. This consistency of HTST-DBN can be attributed to the synergy of TST and DBN architectures, alongside IOA-driven hyper-parameter optimization, which collectively minimize randomness in training. Such stability is crucial for real-world applications, particularly in IoT and 5G/6G scenarios where reproducibility and reliability are as important as accuracy. The extremely low variance values confirm that HTST-DBN is not only more accurate but also more dependable, making it a strong candidate for

deployment in future intelligent network monitoring systems and mission-critical anomaly detection tasks.

Table 6. Variance of different models across 30 independent runs.

Model	Variance			
	CICDDoS2019	**Edge-IoTset**	**5G-NIDD**	**ToN-IoT**
HTST-DBN	0.00035	0.00046	0.00061	0.00072
TST	2.18563	2.93214	3.63215	3.96321
BERT	2.82145	3.53214	4.24136	4.86879
DBN	3.28632	4.05214	5.12546	5.90745
DRL	4.18963	4.96523	6.08632	6.96325
CNN	6.02145	6.90145	7.17523	7.84523
RF	9.41230	9.72365	10.18532	10.79632

Table 7 shows the outcomes of statistical *t*-tests conducted between the proposed HTST-DBN and each baseline model across all datasets. The *p*-values for every comparison are well below the threshold of 0.01, confirming that the performance differences observed are statistically significant. The "Results" column consistently reports "Significant," indicating that the improvements achieved by HTST-DBN are not due to random variation but are statistically reliable across multiple datasets. The findings reinforce that the superior accuracy, recall, and AUC demonstrated in earlier tables are supported by rigorous statistical validation. For example, even in challenging datasets like ToN-IoT and 5G-NIDD, the *p*-values remain as low as 0.0002 or smaller, showing strong confidence in HTST-DBN's advantage over baselines. This robustness across diverse environments highlights the reliability of the proposed architecture for real-world deployment. By passing significance testing at the 0.01 level, HTST-DBN proves not only to be more effective but also consistently better than existing models in a statistically meaningful sense, further solidifying its potential for anomaly detection in next-generation communication networks.

Table 7. Statistical *t*-test results comparing HTST-DBN with others at a 0.01 significance level.

Model	Statistical *t*-Tests							
	CICDDoS2019		**Edge-IoTset**		**5G-NIDD**		**ToN-IoT**	
	***p*-Value**	**Results**	***p*-Value**	**Results**	***p*-Value**	**Results**	***p*-Value**	**Results**
HTST-DBN vs. TST	0.0009	Significant	0.0008	Significant	0.0007	Significant	0.0005	Significant
HTST-DBN vs. BERT	0.0008	Significant	0.0006	Significant	0.0008	Significant	0.0004	Significant
HTST-DBN vs. DBN	0.0006	Significant	0.0005	Significant	0.0007	Significant	0.0003	Significant
HTST-DBN vs. DRL	0.0002	Significant	0.0001	Significant	0.0003	Significant	0.0001	Significant
HTST-DBN vs. CNN	0.00009	Significant	0.00008	Significant	0.00005	Significant	0.00003	Significant
HTST-DBN vs. RF	0.000008	Significant	0.000006	Significant	0.000004	Significant	0.000002	Significant

Table 8 shows the execution time required for each model to reach different training termination thresholds. This experiment measures the computational efficiency of the models by observing how quickly they achieve increasingly strict error levels during training. Such an evaluation provides practical insight into the scalability of models, especially when considering real-world deployments where both accuracy and computational cost are critical. From the results, HTST-DBN demonstrates a clear advantage in efficiency, requiring only 42 s to reach RMSE < 15 and 309 s to achieve RMSE < 3, a threshold that no baseline model was able to reach. By contrast, TST and BERT needed 849 s and 962 s, respectively, just to reach RMSE < 5, while CNN and DBN converged much more slowly, and RF failed to meet the stricter thresholds. These outcomes highlight that the hybrid integration of TST and DBN, coupled with IOA optimization, not only enhances accuracy and stability but

also ensures faster convergence, reducing computational overhead significantly compared to other deep learning and classical methods.

Table 8. Comparison of runtime for CICDDoS2019 under different RMSE-based stopping criteria.

Proposed Methods	Runtime (s)			
	RMSE < 15	RMSE < 10	RMSE < 5	RMSE < 3
HTST-DBN	42	98	176	309
TST	132	392	849	-
BERT	148	401	962	-
DBN	160	419	-	-
DRL	155	493	-	-
CNN	179	625	-	-
RF	211	-	-	-

Table 9 presents the inference latency results for the proposed HTST-DBN model and comparative baselines, measured in milliseconds per sample. Each value represents the average time required by a fully trained model to infer a single test instance on a standard CPU setup, emulating edge-level computational resources. Lower latency values indicate faster real-time inference and, therefore, higher suitability for on-device deployment in IoT or edge networks. Across all datasets, the proposed HTST-DBN maintains an average latency between 6.5 ms and 8.7 ms, which is well below the 10 ms threshold typically considered acceptable for real-time IoT detection. The model exhibits slightly higher latency than shallower networks such as TST and RF but remains significantly faster than heavy architectures like BERT and DRL, which require between 7.1 ms and 9.6 ms on average. These results confirm that the hybrid integration of TST and DBN provides a favorable balance between computational complexity and inference efficiency, enabling high detection accuracy without excessive runtime cost. From a broader perspective, the latency difference across datasets reflects the varying input dimensionality and sequence lengths of the respective benchmarks. For instance, CICDDoS2019 and ToN-IoT (both containing longer traffic flows) result in slightly increased inference times, whereas 5G-NIDD yields the lowest latency due to its compact signaling structure.

Table 9. Average inference latency (in milliseconds) of all benchmarked models across four datasets.

Model	Inference Latency (ms)			
	CICDDoS2019	Edge-IoTset	5G-NIDD	ToN-IoT
HTST-DBN	8.2	7.2	6.5	8.7
TST	6.3	5.5	5.1	7.2
BERT	9.1	7.9	6.8	9.4
DBN	6.2	4.3	4.8	7.3
DRL	8.3	7.4	7.1	8.9
CNN	9.4	8.6	7.2	9.6
RF	5.8	5.2	5.3	7.1

Taken together with the variance (Table 6) and *t*-test results (Table 7), the runtime analysis (Table 8) and inference latency (Table 9) underscores the overall strength of HTST-DBN. The model is not only more accurate and stable, but also statistically validated and computationally efficient. These complementary dimensions confirm that HTST-DBN is well-suited for real-world anomaly detection in mobile and IoT networks, where high accuracy, reliability, and efficiency must coexist. By excelling across performance, stability, and execution time, the proposed model demonstrates its readiness for practical deployment in next-generation communication infrastructures.

The principles underlying the proposed framework can be seamlessly extended to circuit and system optimization tasks in EDA. In complex design spaces, achieving optimal performance requires balancing multiple interdependent parameters such as device sizing, bias conditions, and layout geometries. Symmetry in these domains typically ensures stable and predictable operation, whereas controlled asymmetry can compensate process variations or improve critical performance metrics such as gain, bandwidth, and power efficiency. The HTST-DBN provides an adaptive, data-driven mechanism to learn and evaluate these symmetric and asymmetric behaviors, enabling a structured understanding of design regularities and deviations.

From a modeling standpoint, the TST and DBN can process transient simulation data (such as voltage, current, and noise waveforms) to extract temporal dependencies and latent structural correlations. Balanced circuit blocks like differential pairs or current mirrors exhibit symmetric temporal patterns, while asymmetries often signify mismatch, nonlinearity, or instability. By capturing these relationships, the hybrid model can identify beneficial asymmetries that enhance circuit robustness while suppressing those that degrade reliability. At the optimization level, the IOA complements this learning process through controlled symmetry breaking. Hard symmetry constraints maintain structural balance, whereas soft penalties allow limited deviation when advantageous. The seasonal-migration operator dynamically alternates between symmetry preservation and asymmetry introduction. Applied to differential or mirrored topologies, this mechanism enables IOA to disrupt ineffective regularities and reinforce optimal, balanced structures. In essence, the combined HTST-DBN framework provides a unified, symmetry-aware optimization paradigm that bridges intelligent anomaly detection in communication networks with adaptive design automation in circuit and system engineering.

5. Conclusions

This paper presents a novel hybrid framework, HTST-DBN, designed to address the challenges of anomaly detection in complex and heterogeneous mobile communication networks. By integrating the sequence modeling capacity of the TST with the deep feature abstraction of the DBN, and further enhancing this architecture through the IOA for hyperparameter optimization, the proposed approach brings together complementary strengths that are rarely combined in the existing literature. This work not only addresses limitations of individual models, such as weak temporal sensitivity or unstable generalization, but also establishes a new pathway for building scalable and adaptive anomaly detection solutions suitable for dynamic 5G and emerging 6G environments.

Across all four datasets, the proposed HTST-DBN consistently outperformed baseline models in terms of classification accuracy, recall, precision, and AUC. On CICDDoS2019 and Edge-IoTset, the framework achieved near-perfect results, with accuracies above 99.6%, recall and precision exceeding 99.7%, and AUC values close to 0.999, clearly surpassing TST, BERT, and DBN which plateaued between 88 and 92% accuracy. Even on the more complex 5G-NIDD and ToN-IoT datasets, where noise and heterogeneity make anomaly detection more difficult, HTST-DBN maintained accuracies above 99.4%, while baselines like CNN and RF struggled to exceed 82–85%. These results underline the hybrid model's superior ability to generalize across diverse environments, while also minimizing missed detections—an aspect particularly evident in its consistently higher recall.

Beyond accuracy-based metrics, the framework also demonstrated stronger training efficiency and robustness. RMSE curves showed that HTST-DBN converged smoothly within 70–90 epochs, far faster than baselines that required over 200 epochs, while execution time analysis revealed that it reached strict RMSE thresholds up to 3.0 in less than 310 s, a feat not achieved by competing methods. Variance analysis across 30 independent runs

confirmed its stability, with values close to zero, whereas traditional baselines exhibited fluctuations an order of magnitude higher. Finally, statistical validation using the *t*-test at a 0.01 significance level confirmed that improvements were not random, but rather were consistently significant across datasets. Together, these findings establish HTST-DBN as an accurate and efficient anomaly detection framework, demonstrating robustness, reliability, and scalability for practical use.

The overall outcome of this work shows that improving anomaly detection in next-generation mobile communication networks requires more than small gains in accuracy. It demands architectures that integrate temporal awareness, deep feature abstraction, and adaptive optimization. These capabilities need to work together in a unified framework to address the complexity of modern networks. By combining these elements, the proposed framework shows that it is possible to design models that not only handle the complexity, heterogeneity, and noise inherent in modern network environments, but also remain computationally feasible and statistically reliable. This finding reinforces the idea that robust anomaly detection is a cornerstone for securing 5G and future 6G infrastructures, where resilience, adaptability, and scalability are essential. Ultimately, this study provides evidence that hybrid, optimization-driven deep learning approaches can form a practical foundation for building intelligent, secure, and sustainable communication systems in real-world deployments.

Future research can extend this work in several promising directions. From the problem perspective, anomaly detection in mobile and IoT networks is expected to become even more complex in 6G and beyond, with the integration of ultra-dense device deployments, edge–cloud cooperation, and heterogeneous traffic patterns. Addressing these environments will require adapting the proposed framework to operate in distributed and federated settings, where privacy and communication constraints play a key role. Moreover, extending the model to support real-time detection on resource-constrained devices will be crucial for applications such as industrial IoT, autonomous vehicles, and mission-critical communications. From the model perspective, the HTST-DBN can be enriched with interpretability modules to improve explainability for network operators, ensuring that its predictions are not only accurate, but are also transparent and actionable. The optimization process itself can also be enhanced by exploring hybrid or co-evolutionary strategies that combine IOA with other search mechanisms to further refine exploration–exploitation trade-offs. Finally, evaluating the model in live testbeds and integrating it with end-to-end network management systems will help bridge the gap between simulation-based validation and real-world deployment, paving the way for practical adoption in future intelligent communication infrastructures.

Author Contributions: Conceptualization, A.E.O., M.K., F.H.-G. and D.M.; methodology, A.E.O., M.K. and D.M.; software, A.E.O. and F.H.-G.; validation, A.E.O., M.K. and F.H.-G.; formal analysis, A.E.O. and D.M.; investigation, A.E.O., M.K., F.H.-G. and D.M.; resources, D.M.; data curation, M.K. and F.H.-G.; writing—original draft preparation, A.E.O., M.K. and F.H.-G.; writing—review and editing, A.E.O., M.K. and D.M.; visualization, A.E.O., M.K. and F.H.-G.; supervision, D.M.; project administration, D.M.; funding acquisition, D.M. 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. Dauda, A.; Flauzac, O.; Nolot, F. A survey on IoT application architectures. *Sensors* **2024**, *24*, 5320. [CrossRef] [PubMed]
2. 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]
3. Madanian, S.; Chinbat, T.; Subasinghage, M.; Airehrour, D.; Hassandoust, F.; Yongchareon, S. Health IoT threats: Survey of risks and vulnerabilities. *Future Internet* **2024**, *16*, 389. [CrossRef]
4. Kaveh, M.; Ghadi, F.R.; Li, Z.; Yan, Z.; Jäntti, R. Secure backscatter communications through RIS: Modeling and performance. *IEEE Trans. Veh. Technol.* **2025**, *74*, 1–13. [CrossRef]
5. Jäntti, R.; Liao, J.; Zhang, T.; Xie, B.; Sheikh, M.U.; Ruttik, K.; Phan, G.; Phan-Huy, D.T.; Hassan, A. Integration of backscatter-based ambient Internet of Things to cellular communication systems. *IEEE Commun. Stand. Mag.* **2025**, *9*, 1–8. [CrossRef]
6. Mondal, S.; Bepari, D.; Chandra, A.; Singh, K.; Li, C.P.; Ding, Z. A comprehensive survey on NOMA-based backscatter communication for IoT applications. *IEEE Internet Things J.* **2025**, *12*, 18929–18953. [CrossRef]
7. Ghadi, F.R.; Kaveh, M.; Wong, K.K.; Zhang, Y. Performance analysis of FAS-aided backscatter communications. *IEEE Wirel. Commun. Lett.* **2024**, *13*, 2412–2416. [CrossRef]
8. Mutambik, I. Sustainable IoT-enabled parking management: A multiagent simulation framework for smart urban mobility. *Sustainability* **2025**, *17*, 6382. [CrossRef]
9. Ghadi, F.R.; Kaveh, M.; Martín, D. Performance analysis of RIS/STAR-IOS-aided V2V NOMA/OMA communications over composite fading channels. *IEEE Trans. Intell. Veh.* **2023**, *9*, 279–286. [CrossRef]
10. Liu, J.; Liu, Y.; Gao, K.; Wang, L. Generative edge intelligence for IoT-assisted vehicle accident detection: Challenges and prospects. *IEEE Internet Things Mag.* **2024**, *7*, 50–54. [CrossRef]
11. Ghadi, F.R.; Kaveh, M.; Wong, K.K.; Martín, D. Physical layer security performance of cooperative dual-RIS-aided V2V NOMA communications. *IEEE Syst. J.* **2024**, *18*, 2074–2084. [CrossRef]
12. Dritsas, E.; Trigka, M. A survey on the applications of cloud computing in the industrial Internet of Things. *Big Data Cogn. Comput.* **2025**, *9*, 44. [CrossRef]
13. Tera, S.P.; Chinthaginjala, R.; Pau, G.; Kim, T.H. Towards 6G: An overview of the next generation of intelligent network connectivity. *IEEE Access* **2024**, *13*, 925–961. [CrossRef]
14. Mishra, R.; Mishra, A. Current research on Internet of Things (IoT) security protocols: A survey. *Comput. Secur.* **2025**, *151*, 104310. [CrossRef]
15. 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]
16. Adam, M.; Hammoudeh, M.; Alrawashdeh, R.; Alsulaimy, B. A survey on security, privacy, trust, and architectural challenges in IoT systems. *IEEE Access* **2024**, *12*, 57128–57149. [CrossRef]
17. Kaveh, M.; Mosavi, M.R.; Martín, 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]
18. 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]
19. Lotfy, A.; Kaveh, M.; Martín, D.; Mosavi, M.R. An efficient design of Anderson PUF by utilization of the Xilinx primitives in the SLICEM. *IEEE Access* **2021**, *9*, 23025–23034. [CrossRef]
20. Kaveh, M.; Ghadi, F.R.; Zhang, Y.; Yan, Z.; Jäntti, R. Voltage profile-driven physical layer authentication for RIS-aided backscattering tag-to-tag networks. *IEEE Internet Things J.* **2025**. [CrossRef]
21. Illi, E.; Qaraqe, M.; Althunibat, S.; Alhasanat, A.; Alsafasfeh, M.; De Ree, M.; Mantas, G.; Rodriguez, J.; Aman, W.; Al-Kuwari, S.; et al. Physical layer security for authentication, confidentiality, and malicious node detection: A paradigm shift in securing IoT networks. *IEEE Commun. Surv. Tutor.* **2023**, *26*, 347–388.
22. Kaveh, M.; Rostami Ghadi, F.; Jäntti, R.; Yan, Z. Secrecy performance analysis of backscatter communications with side information. *Sensors* **2023**, *23*, 8358. [CrossRef]
23. De Keersmaeker, F.; Cao, Y.; Ndonda, G.K.; Sadre, R. A survey of public IoT datasets for network security research. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1808–1840. [CrossRef]
24. Sun, P.; Shen, S.; Wan, Y.; Wu, Z.; Fang, Z.; Gao, X.Z. A Survey of IoT Privacy Security: Architecture, Technology, Challenges, and Trends. *IEEE Internet Things J.* **2024**, *11*, 34567–34591. [CrossRef]
25. Fard, S.S.; Kaveh, M.; Mosavi, M.R.; Ko, S.B. An efficient modeling attack for breaking the security of XOR-Arbiter PUFs by using the fully connected and long-short term memory. *Microprocess. Microsyst.* **2022**, *94*, 104667. [CrossRef]
26. Dritsas, E.; Trigka, M. A survey on cybersecurity in IoT. *Future Internet* **2025**, *17*, 30. [CrossRef]

27. Kikissagbe, B.R.; Adda, M. Machine learning-based intrusion detection methods in IoT systems: A comprehensive review. *Electronics* **2024**, *13*, 3601. [CrossRef]
28. Al-Haija, Q.A.; Droos, A. A comprehensive survey on deep learning-based intrusion detection systems in Internet of Things (IoT). *Expert Syst.* **2025**, *42*, e13726. [CrossRef]
29. Abreu, R.; Simão, E.; Serôdio, C.; Branco, F.; Valente, A. Enhancing IoT security in vehicles: A comprehensive review of AI-driven solutions for cyber-threat detection. *AI* **2024**, *5*, 2279–2299.
30. Momand, A.; Jan, S.U.; Ramzan, N. ABCNN-IDS: Attention-based convolutional neural network for intrusion detection in IoT networks. *Wirel. Pers. Commun.* **2024**, *136*, 1981–2003.
31. Almotairi, A.; Atawneh, S.; Khashan, O.A.; Khafajah, N.M. Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. *Syst. Sci. Control Eng.* **2024**, *12*, 2321381. [CrossRef]
32. Akgun, D.; Hizal, S.; Cavusoglu, U. A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. *Comput. Secur.* **2022**, *118*, 102748. [CrossRef]
33. Ramzan, M.; Shoaib, M.; Altaf, A.; Arshad, S.; Iqbal, F.; Castilla, Á.K.; Ashraf, I. Distributed denial of service attack detection in network traffic using deep learning algorithm. *Sensors* **2023**, *23*, 8642. [CrossRef]
34. Lopez-Martin, M.; Sanchez-Esguevillas, A.; Arribas, J.I.; Carro, B. Network intrusion detection based on extended RBF neural network with offline reinforcement learning. *IEEE Access* **2021**, *9*, 153153–153170. [CrossRef]
35. Li, J.; Othman, M.S.; Chen, H.; Yusuf, L.M. Optimizing IoT intrusion detection system: Feature selection versus feature extraction in machine learning. *J. Big Data* **2024**, *11*, 36. [CrossRef]
36. Harshdeep, K.; Sumalatha, K.; Mathur, R. DeepTransIDS: Transformer-based deep learning model for detecting DDoS attacks on 5G NIDD. *Results Eng.* **2025**, *26*, 104826. [CrossRef]
37. Farzaneh, B.; Shahriar, N.; Al Muktadir, A.H.; Towhid, M.S.; Khosravani, M.S. DTL-5G: Deep transfer learning-based DDoS attack detection in 5G and beyond networks. *Comput. Commun.* **2024**, *228*, 107927.
38. Sakr, H.A.; El-Afifi, M.I.; El-Mowafy, M.A.; Ibrahim, H.M. Detecting DDoS threats in IoT-driven 6G-energy hubs networks using machine learning algorithms. *Discov. Appl. Sci.* **2025**, *7*, 1002.
39. Toralkar, P.; Mainalli, K.; Allagi, S.; Debnath, S.K.; Bagchi, S.; Leong, W.Y.; Khan, M.N.A. Enhanced intrusion detection with advanced deep features and ensemble classifier techniques. *SN Comput. Sci.* **2025**, *6*, 381. [CrossRef]
40. Barona, R.; Baburaj, E. An efficient DDoS attack detection and categorization using adolescent identity search-based weighted SVM model. *Peer-to-Peer Netw. Appl.* **2023**, *16*, 1227–1241.
41. Lopes, I.O.; Zou, D.; Abdulqadder, I.H.; Akbar, S.; Li, Z.; Ruambo, F.; Pereira, W. Network intrusion detection based on the temporal convolutional model. *Comput. Secur.* **2023**, *135*, 103465. [CrossRef]
42. Mall, R.; Abhishek, K.; Shankar, A.; Kumar, A. Stacking ensemble approach for DDoS attack detection in software-defined cyber–physical systems. *Comput. Electr. Eng.* **2023**, *107*, 108635. [CrossRef]
43. Sadhwani, S.; Mathur, A.; Muthalagu, R.; Pawar, P.M. 5G-SIID: An intelligent hybrid DDoS intrusion detector for 5G IoT networks. *Int. J. Mach. Learn. Cybern.* **2025**, *16*, 1243–1263. [CrossRef]
44. Moubayed, A. A complete EDA and DL pipeline for softwarized 5G network intrusion detection. *Future Internet* **2024**, *16*, 331. [CrossRef]
45. Ayad, A.G.; Sakr, N.A.; Hikal, N.A. A hybrid approach for efficient feature selection in anomaly intrusion detection for IoT networks. *J. Supercomput.* **2024**, *80*, 26942–26984. [CrossRef]
46. Jin, Y.; Hou, L.; Chen, Y. A time series transformer based method for the rotating machinery fault diagnosis. *Neurocomputing* **2022**, *494*, 379–395. [CrossRef]
47. Kim, J.; Kang, H.; Kang, P. Time-series anomaly detection with stacked Transformer representations and 1D convolutional network. *Eng. Appl. Artif. Intell.* **2023**, *120*, 105964. [CrossRef]
48. Li, Z.; Zhang, X.; Dong, Z. TSF-transformer: A time series forecasting model for exhaust gas emission using transformer. *Appl. Intell.* **2023**, *53*, 17211–17225. [CrossRef] [PubMed]
49. Keshun, Y.; Yingkui, G.; Yanghui, L.; Yajun, W. A Novel Physical Constraint-Guided Quadratic Neural Networks for Interpretable Bearing Fault Diagnosis under Zero-Fault Sample. *Nondestruct. Test. Eval.* **2025**, *40*, 1–31. [CrossRef]
50. Chen, Z.; Yang, J.; Chen, L.; Li, F.; Feng, Z.; Jia, L.; Li, P. RailVoxelDet: A Lightweight 3-D Object Detection Method for Railway Transportation Driven by Onboard LiDAR Data. *IEEE Internet Things J.* **2025**, *12*, 37175–37189. [CrossRef]
51. Zheng, R.; Yang, B.; Qian, Y.; Li, H.; Gao, D.; Jiang, L. Joint SOH and RUL estimation for lithium-ion batteries via optimal deep belief network with Bayesian algorithm. *J. Energy Storage* **2025**, *114*, 115891.
52. Savitha, S.; Kannan, A.R.; Logeswaran, K. Augmenting cardiovascular disease prediction through CWCF integration leveraging Harris Hawks search in deep belief networks. *Cogn. Comput.* **2025**, *17*, 52. [CrossRef]
53. Meng, S.; Shi, Z.; Li, G.; Peng, M.; Liu, L.; Zheng, H.; Zhou, C. A novel deep learning framework for landslide susceptibility assessment using improved deep belief networks with the intelligent optimization algorithm. *Comput. Geotech.* **2024**, *167*, 106106. [CrossRef]

54. Luo, J.; Zhao, C.; Chen, Q.; Li, G. Using deep belief network to construct the agricultural information system based on Internet of Things. *J. Supercomput.* **2022**, *78*, 379–405.
55. Kaveh, M.; Mesgari, M.S.; Saeidian, B. Orchard algorithm (OA): A new meta-heuristic algorithm for solving discrete and continuous optimization problems. *Math. Comput. Simul.* **2023**, *208*, 95–135. [CrossRef]