Neural Computing and Applications (2025) 37:2455–2477
https://doi.org/10.1007/s00521-024-10759-1

ORIGINAL ARTICLE

TransNAS-TSAD: harnessing transformers for multi-objective neural architecture search in time series anomaly detection

Ijaz Ul Haq[1] · Byung Suk Lee[1] · Donna M. Rizzo[2]

Received: 14 March 2024 / Accepted: 28 October 2024 / Published online: 5 December 2024

Abstract

The surge in real-time data collection across various industries has underscored the need for advanced anomaly detection in both univariate and multivariate time series data. This paper introduces TransNAS-TSAD, a framework that synergizes the transformer architecture with neural architecture search (NAS), enhanced through NSGA-II algorithm optimization. This approach effectively tackles the complexities of time series data, balancing computational efficiency with detection accuracy. Our evaluation reveals that TransNAS-TSAD surpasses conventional anomaly detection models due to its tailored architectural adaptability and the efficient exploration of complex search spaces, leading to marked improvements in diverse data scenarios. We also introduce the Efficiency-accuracy-complexity score (EACS) as a composite metric that balances accuracy, computational efficiency, and model complexity, providing a comprehensive assessment of model performance. TransNAS-TSAD sets a new benchmark in time series anomaly detection, offering a versatile, efficient solution for complex real-world applications. This research highlights TransNAS-TSAD's potential across a wide range of industry applications and paves the way for future developments in the field.

Keywords Neural architecture search · Time series anomaly detection · transformers · Multi-objective optimization

1 Introduction

The ubiquity of time series data across various sectors, ranging from finance [1] and healthcare [2] to infrastructure [3] and manufacturing [4], underscores its pivotal role in modern analytics. These type of data are instrumental in identifying the patterns, dependencies, and anomalies indicative of significant shifts in system behaviors or the emergence of critical issues [5, 6].

Traditionally, statistical methods have been the cornerstone for anomaly detection in time series data, and are renowned for their robust mathematical frameworks [7–9]. However, the advent of Big Data, characterized by its significant volume, velocity, and variety, has revealed the limitations of these traditional methods, including high false positive rates and missed detections [10, 11].

In response, deep learning has introduced a paradigm shift, offering enhanced adaptability and accuracy in anomaly detection tasks [12–14]. This benefits usually come at a high cost on the computational efficiency due to the high complexity of the model. Notably, however, the transformer architecture, with its self-attention mechanism, has emerged as a groundbreaking development in enhancing not only the accuracy but also the computational efficiency. This enabled it to make significant strides not only in natural language processing but also in time series analysis [15–17].

Despite the advantages offered by transformer models, their application to time series anomaly detection necessitates a more dynamic and adaptable framework for fine-tuning their structures and parameters for optimal performance (balancing accuracy and computational efficiency) [18]. Our model, TransNAS-TSAD, combines the

✉ Ijaz Ul Haq
ihaq@uvm.edu

Byung Suk Lee
bslee@uvm.edu

Donna M. Rizzo
drizzo@uvm.edu

[1] Department of Computer Science, University of Vermont, Burlington, VT 05405, USA

[2] Department of Civil and Environmental Engineering, University of Vermont, Burlington, VT 05405, USA

architectural strengths of transformers (e.g., TranAD) with the optimization strategy of a neural architecture search (NAS) to systematically explore a huge number of architectural configurations. The latter leverages the multi-objective optimization capabilities of the NSGA-II algorithm [19, 20], enabling an efficient exploration of complex search spaces [21, 22]. This represents a substantial evolution of the TranAD's static architecture, facilitating a balance between accuracy and computational efficiency.

Adapting anomaly detection models to time series data is inherently complex, especially when addressing multivariate series that present high-dimensional challenges and necessitate sophisticated model architectures [23, 24]. The selective attention capabilities of the transformer model offer a compelling solution, provided the architectures are meticulously tailored to the specific research questions and dataset characteristics [25].

The convergence of NAS with multi-objective optimization algorithms like NSGA-II marks a strategic evolution in automated model design. This approach provides a methodical pathway to discovering optimal architectures that consider multiple, often conflicting, objectives, thereby enhancing the adaptability and effectiveness of anomaly detection models [26, 27].

Our research is grounded on three fundamental principles designed to tackle the inherent challenges of time series anomaly detection. First, we emphasize the necessity of developing specialized models tailored to the unique temporal resolution and dimensional characteristics of time series data, ensuring they are finely attuned to the intricacies of temporal patterns. Second, we aim to maximize the utility of transformer architectures, exploiting their advanced capabilities for deep and nuanced interpretation of temporal data, which is critical for identifying subtle anomalies. Lastly, strategic optimization through neural architecture search (NAS), particularly leveraging the NSGA-II algorithm, forms the cornerstone of our methodology. This approach allows us to systematically refine and perfect our models, ensuring they are not only highly effective but also optimized for the specific demands of the datasets they analyze. Together, these principles guide our pursuit of creating models that are both innovative and highly adapted to the complex domain of time series anomaly detection. Our contributions are delineated as follows:

- We introduce a novel approach that applies NAS with NSGA-II optimization to specifically fine-tune transformer architectures for time series anomaly detection, showcasing NAS's adaptability in specialized domains.
- Our experimentation identifies key architectural configurations that elevate detection performance, highlighting the importance of architectural fine-tuning in enhancing model efficacy.
- We demonstrate substantial performance improvements across diverse datasets, proving our optimized models' superiority over existing methods and illustrating the benefits of multi-objective optimization for anomaly detection.
- We strike an optimal balance between analytical accuracy and computational efficiency, the two performance factors important in our work, presenting models that excel in real-time anomaly detection and are feasible for practical deployment, emphasizing efficiency in model design.
- The synergy between transformers' structural strengths and NAS's adaptive optimization creates uniquely effective models for time series anomaly detection, marking a significant advancement by integrating architectural innovation with strategic optimization.

2 Related work

The quest to detect and understand anomalies within time series data spans various domains including, but not limited to, finance [1], healthcare [2], infrastructure [3], and manufacturing [4]. This widespread interest underscores the fundamental role of anomaly detection in predictive maintenance, fraud detection, and system health monitoring, among other critical applications [5, 6]. In finance, for example, anomaly detection can signal fraudulent transactions or market manipulations, while in healthcare, it may indicate abnormal patient conditions that require immediate attention [1].

Initially, the field of time series anomaly detection heavily relied on statistical methods, grounded in robust mathematical frameworks, that have provided deep insights into data patterns and anomalies [8, 9]. These methods, while effective for simpler datasets, often assume specific data distributions or employ distance-based metrics to identify outliers. However, their application to the high-dimensional, complex nature of multivariate time series data reveals significant limitations. The primary challenge lies not merely in capturing the correlations within the data, but in effectively managing the intricacies associated with multi-correlated variables. These complexities can lead to an increased rate of false positives and missed detections, as traditional methods struggle to adequately interpret the nuanced interactions and dependencies that characterize modern datasets [28, 29].

The emergence of deep learning models, and their ability to learn high-level representations from massive amounts of data, have shown remarkable success in

detecting anomalies in time series data [13]. These models excel in identifying deviations from frequently occurring patterns, significantly reducing false positives and improving detection accuracy [30–32]. A deeper exploration of machine learning models, including specific architectures and their successes across various applications, would provide valuable insights [26].

Among the deep learning architectures that have gained prominence, the transformer architecture introduced by Vaswani et al. [15] has revolutionized how models process sequential data. Unlike its predecessors, such as recurrent neural networks (RNNs) and long short-term memory (LSTMs), the transformer leverages a self-attention mechanism, enabling it to capture long-range dependencies and temporal patterns in data without being hindered by the sequential processing bottleneck [33–37]. Its application in time series anomaly detection is particularly promising, offering a new perspective on modeling temporal anomalies. However, this area remains relatively unexplored, with significant potential for optimization and adaptation to the unique characteristics of time series data across different domains [7].

The neural architecture search (NAS) has emerged as a transformative approach in automating the design of neural network architectures [22, 38, 39]. By systematically exploring a vast space of architectural configurations, NAS aims to identify models that achieve optimal performance for a given task. This automation is crucial in deep learning, where the design and tuning of models are both resource-intensive and require specialized knowledge [26, 40, 41]. Recent advancements in NAS have focused on improving its efficiency, reducing the computational resources required for the search process, and making state-of-the-art model architectures accessible for a broader range of applications [38, 42].

Incorporating multi-objective optimization into NAS, using algorithms like NSGA-II, represents a significant leap forward [20, 41]. For time series anomaly detection, where models must process large volumes of data accurately and with computational efficiency, the ability to find a nuanced balance between these competing objectives is crucial [38, 43, 44]. Despite the clear advantages, the application of multi-objective optimization in NAS for anomaly detection in time series is ripe for exploration and offers vast potential for groundbreaking research [26].

Our work seeks to harness the strengths of neural architecture search (NAS) and transformer models to directly address the unique challenges associated with temporal data, crafting models that are finely tuned for anomaly detection in time series. This integration aims to develop architectures that excel in both performance and efficiency, leveraging the latest advancements in the field [45]. Given the nascent stage of combining NAS with transformer models for this purpose, our research not only contributes to the existing body of knowledge but also opens avenues for future exploration and innovation in anomaly detection methodologies [46]. By situating our study within the context of ongoing research efforts and emerging trends, we aim to advance the state of the art in anomaly detection within time series data and contribute meaningfully to the broader discourse in this rapidly evolving field.

3 Methodology

In this section, we delineate the comprehensive methodology employed in our study, designed to effectively detect and analyze anomalies in multivariate time series data using the TransNAS-TSAD framework. We commence with a clear definition of the problem, setting the stage for the subsequent methodological steps. This is followed by an in-depth discussion on data refinement techniques, ensuring that the data are appropriately pre-processed for our analysis. Next, we present the intricacies of our transformer architecture, highlighting its design and capabilities. A critical component of our approach is the neural architecture search (NAS), which employs the NSGA-II algorithm for optimizing the transformer model. That description is supplemented by an overview of the evolutionary process that iteratively refines our model architecture. We then transition to discussing advanced anomaly detection techniques, integrating adversarial elements into our model. The section culminates with a description of how we harness the full potential of TransNAS-TSAD for practical anomaly detection applications, illustrating the real-world implications of our research.

3.1 Problem definition

Our research pioneers the application of a multi-objective approach to anomaly detection in multivariate time series data, utilizing transformer models. The problem is defined as follows.

Given a sequence of multivariate time series data $S = \{x_1, x_2, \ldots, x_T\}$, where each point x_t is timestamped t and resides in an m-dimensional space ($x_t \in \mathbb{R}^m$), our aims are to perform:

1. *Anomaly detection in time series*: Using a training time series S and a test series S' of length T' with similar modalities, we seek to detect anomalies by predicting a binary sequence $Y = \{y_1, y_2, \ldots, y_{T'}\}$, where y_t identifies anomalies at timestamp t in S' (1 indicates an anomaly).

2. *Anomaly component analysis*: Our goal extends to identifying anomalous components within each data point of the series, generating a detailed prediction sequence $Y = \{y_1, y_2, \ldots, y_{T'}\}$, where each y_t pinpoints the specific anomalous dimensions at timestamp t.

Multi-objective transformer model optimization

At the core of our methodology lies the NSGA-II-based neural architecture search (NAS), which dynamically optimizes transformer architectures specifically for anomaly detection in time series data. Unlike models such as TranAD[16], which employs a static architecture across all datasets, we propose a novel approach that adapts the model's structure to suit each dataset's characteristics. This dynamic approach ensures that the architecture is optimized for both accuracy and efficiency in each case. Our multi-objective optimization focuses on:

- *Maximizing anomaly detection accuracy*: Unlike static models, we tune the transformer architecture dynamically to achieve high F1 scores across datasets. This flexibility allows our model to adapt to different anomaly patterns, enhancing detection accuracy.
- *Minimizing architectural complexity*: Simultaneously, we optimize the model's size and computational complexity, ensuring efficiency in deployment. This aspect is critical for practical applications, as our method achieves high detection accuracy without the computational overhead typically associated with transformer models.

Our NAS-based approach, combined with NSGA-II, offers a systematic way to explore architectural configurations, identifying the optimal balance between accuracy and efficiency, which is a significant advancement over previous static models.

Data processing and model training

Prior to model training and classification, the time series data undergo a preprocessing phase where normalization is applied to ensure uniformity of scale across all series components. In this study, we employed min-max normalization, which is widely used in time series data processing due to its ability to map values to a fixed range, typically [0,1], eliminating bias from variables with larger magnitudes and ensuring stability during training. This technique has been shown to perform well in time series forecasting models, particularly in neural networks like LSTMs and transformers [47, 48]. The normalization process is defined as follows:

$$x_t \leftarrow \frac{x_t - \min(S)}{\max(S) - \min(S) + \epsilon},$$

where $\min(S)$ and $\max(S)$ denote the component-wise minimum and maximum values observed in the training and testing time series data S, respectively. The term ϵ is a small constant introduced to prevent division by zero, ensuring numerical stability.

While alternative normalization techniques, such as z-score normalization and robust scaling, exist, min-max normalization was selected for this study because it provides a simple, yet effective, approach to scaling time series data without assuming Gaussian distribution [49]. This method also ensures that features are bounded within a consistent range, which is beneficial for transformer-based models that rely on stable gradients during optimization.

Subsequently, the time series is transformed into a set of overlapping windows, which serve as the input to our model. A context window of size K is defined preceding each data point x_t, resulting in:

$$W_t = \{x_{t-K+1}, \ldots, x_t\},$$

where boundary cases are handled by replicating the first available data point to fill the window, maintaining a fixed size of K. This approach encapsulates the temporal dependencies inherent in time series data.

Anomaly scoring and thresholding

Our model assigns an anomaly score to each context window W_t, rather than providing direct binary labels. This score is derived from the reconstruction error of the window, where O_t denotes the reconstructed output corresponding to W_t. A **dynamic threshold** D is established based on the score distribution from training data, which helps differentiate between normal and anomalous windows.

Unlike traditional fixed-threshold methods commonly used in anomaly detection, we employ a dynamic thresholding mechanism known as modified Peaks-Over-Threshold (mPOT). This method allows the threshold to adapt to variations in the data over time, thereby improving robustness in scenarios with fluctuating data patterns. The mPOT approach integrates recent deviation statistics, ensuring that the model remains responsive to real-time data trends, which reduces the risk of false positives or missed detections in rapidly changing environments.

Additionally, our neural architecture search (NAS) framework ensures that the thresholding mechanism is optimized for a wide range of datasets, balancing accuracy and computational efficiency. Through the iterative self-adversarial phase described in Sect. 3.5, our framework continuously refines the anomaly scores across multiple iterations, further enhancing the detection capability by dynamically adjusting the threshold D. This iterative refinement is a significant improvement over static methods like those used in models such as TranAD.

By combining mPOT with adversarial refinement and adaptive thresholding, we achieve a more flexible and

responsive anomaly detection system that can handle evolving data patterns with greater precision.

3.2 TransNAS-TSAD transformer architecture

In the TransNAS-TSAD framework, the transformer architecture is specially adapted for time series anomaly detection. While based on the standard transformer architecture, TransNAS-TSAD introduces flexibility in its configuration to dynamically align with the specific characteristics of the time series data. This is achieved through the neural architecture search (NAS), which optimizes the transformer model's key components for each dataset rather than relying on a fixed architecture, making our approach more adaptable compared to static models like TranAD.

Data augmentation mechanisms within the model: TransNAS-TSAD integrates specific data augmentation techniques into the model architecture, which are dynamically selected during training, based on the NAS optimization. These techniques include:

- *Gaussian noise augmentation (Optional)*: Adds Gaussian noise to the input sequence to increase variability and robustness during training.
- *Time warping & time masking augmentation (Optional)*: Applies time warping or masking to improve the model's generalization to unseen time series patterns, especially in datasets with irregular time intervals.

Encoder Operations: Unlike static models, TransNAS-TSAD dynamically configures the encoder based on the specific needs of the dataset. The following operations are part of the encoder, but the neural architecture search (NAS) determines which components are active and their configurations:

- *Linear embedding (Optional)*: This operation transforms the input into a higher-dimensional space through linear mapping. Whether or not this operation is used is dynamically selected during the NAS process.
- *Layer normalization*: This operation normalizes each layer to ensure stability during training and is consistently used in the encoder, but the exact configuration (e.g., type of normalization) is fine-tuned during the architecture search.
- *Positional encoding*: Depending on the dataset characteristics, NAS selects either sinusoidal or Fourier positional encoding to capture temporal dependencies in the data effectively.
- *Multi-head self-attention*: A key feature in TransNAS-TSAD is the multi-head attention mechanism, which is aligned with the number of features in the data, inspired by the TranAD model. It employs the standard scaled-dot product attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_{\text{model}}}}\right)V. \quad (1)$$

The multi-head attention mechanism is formulated as:

$$\text{MultiHeadAtt}(Q, K, V) = \text{Concat}(H_1, H_2, \ldots, H_h), \quad (2)$$

where $H_i = \text{Attention}(Q_i, K_i, V_i)$.

- **Feedforward Neural Networks (FFNs) Post-Attention:** After the attention mechanism, the sequence is passed through a series of feedforward neural networks (FFNs). Unlike in static models, where the number of layers and neurons is fixed, TransNAS-TSAD uses NAS to dynamically select the optimal configuration of FFNs, balancing between model complexity and anomaly detection accuracy. This ensures the FFN layers are tailored to the dataset, making the model both efficient and accurate.

Decoder operations: The decoder in TransNAS-TSAD mirrors the complexity of the encoder and is similarly optimized through NAS. It integrates the encoder's output, applies multi-head self-attention, and utilizes feedforward neural networks. The model can dynamically choose between using a dual sequential decoder setup or an iterative approach, depending on the dataset and the optimal configuration selected by NAS. This flexibility allows TransNAS-TSAD to maximize anomaly detection accuracy while maintaining computational efficiency across different time series datasets.

3.3 Neural architecture search (NAS) framework for TransNAS-TSAD

The NAS framework shown in Fig. 1 is pivotal in achieving our primary objective of effectively detecting anomalies in multivariate time series data. By employing the NSGA-II algorithm, our NAS process systematically navigates the vast architectural landscape of transformer models. This step is essential in identifying an architecture that not only excels in anomaly detection accuracy but also aligns with the computational constraints of practical deployment scenarios. The optimal selection of transformer architectures through NAS directly contributes to enhancing the performance and efficiency of the TransNAS-TSAD framework, thereby fulfilling our goal of developing a robust and adaptable anomaly detection system.

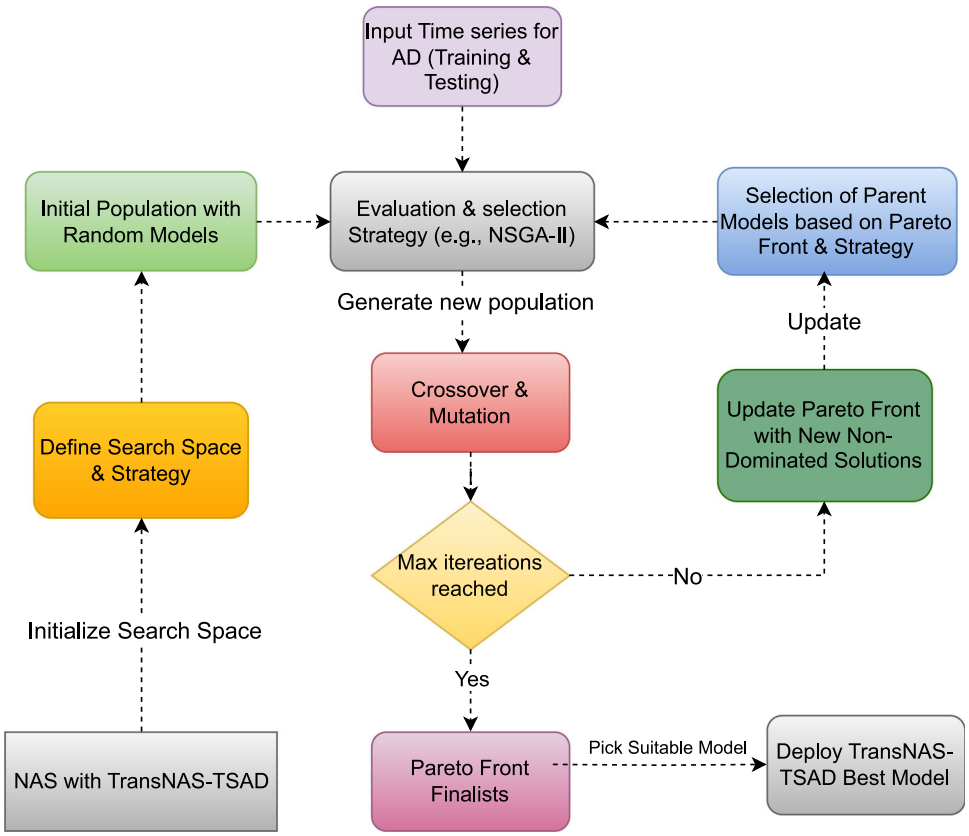

Fig. 1 Workflow of the TransNAS-TSAD process for time series anomaly detection using multi-objective neural architecture search with NSGA-II optimization

Table 1 Neural architecture search space for TransNAS-TSAD

Parameter type	Parameter name	Search space	Description
Training	Learning rate	1×10^{-5} to 1×10^{-1} (log scale)	Rate at which the model learns
	Dropout rate	0.1 to 0.5	Regularization to prevent overfitting
	Batch size	16 to 128 (step of 16)	Number of samples per training step
	Gaussian noise	1×10^{-4} to 1×10^{-1} (log scale)	Noise added for robustness
	Time warping	True, False	Augmentation technique for time series
	Time masking	True, False	Augmentation technique to mask intervals
	Window size	10 to 30	Length of the input sequence window for the model
Architectural	Positional encoding type	Sinusoidal, Fourier	Encoding type for sequence position
	Dimension feedforward	8 to 128 (log scale)	Size of the feedforward network
	Encoder layers	1 to 3	Number of layers in the encoder
	Decoder layers	1 to 3	Number of layers in the decoder
	Activation function	ReLU, Leaky ReLU, Sigmoid, Tanh	Nonlinearity after each layer
	Attention type	Scaled Dot Product	Type of attention mechanism
	Number of attention heads	Equal to feature dimension	Parallel attention layers
	Use linear embedding	True, False	Option to use a linear embedding layer
	Layer normalization	Layer, Batch, Instance	Type of normalization used
	Self-conditioning	True, False	Conditioning strategy for the model
	Number of FFN layers	1 to 3	Layers in the feedforward network
	Phase type	1phase, 2phase, Iterative	Model's reconstruction and refinement strategy

3.3.1 Search space definition

The search space within TransNAS-TSAD, outlined in Table 1, represents a diverse and extensive array of potential model architectures and hyperparameters. This assortment is meticulously tailored toward neural network configurations designed specifically for time-series anomaly detection:

- *Architectural parameters*: These parameters encompass the number and types of layers in the encoder and decoder, optimized attention mechanisms for temporal data analysis, dimensions of feedforward networks, and a variety of positional encoding and normalization methods. Collectively, they enable the model to adapt its architecture for different characteristics of time-series data.
- *Training hyperparameters*: This category includes a broad spectrum of hyperparameters, such as learning rates, batch sizes that are adaptable to varying computational resources, and dropout rates to ensure effective regularization. Additionally, the window size parameter, ranging from 10 to 30, defines the length of the input sequence window, which directly influences how the model processes and learns from temporal patterns. Data augmentation techniques like time warping and masking are also part of the search space, enabling the model to simulate and learn from varied anomaly scenarios effectively.

The comprehensive nature of the search space in TransNAS-TSAD allows for the exploration and optimization of a wide range of models, ensuring that the most suitable architecture and hyperparameter settings are identified for effective anomaly detection in different time-series datasets.

3.3.2 TransNAS-TSAD evaluation strategy

The evolutionary optimization strategy of TransNAS-TSAD entails a rigorous evaluation of models generated with varying architectural and training hyperparameters. The multi-objective evaluation focuses on two critical aspects: the F1 score and the number of model parameters. The F1 score serves as a key indicator of the model's accuracy in anomaly detection, balancing precision and recall, while the number of parameters gauges the model's architectural complexity and computational efficiency. The objective is to identify models that not only demonstrate high proficiency in accurately detecting anomalies (as reflected in a high F1 score) but also maintain a streamlined architecture (evidenced by a lower count of parameters). This dual-criteria assessment ensures the selection of models that are both effective in performance and practical in deployment, aligned with the overarching goal of achieving optimal anomaly detection with computational resourcefulness.

Multi-objective optimization with NSGA-II

TransNAS-TSAD employs the NSGA-II algorithm, a widely recognized approach for balancing the objectives of accuracy and computational efficiency in multi-objective optimization. Our choice of NSGA-II is supported by benchmark studies, such as those presented in the work by Lu et al. [21], which highlight its effectiveness in identifying optimal solutions across various search spaces. This empirical evidence demonstrates NSGA-II as a suitable approach for optimizing the competing demands of our model architecture. The optimization process in TransNAS-TSAD is governed by the following equations:

$$\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})], \tag{3}$$

where $f_1(\mathbf{x})$ is the F1 score and $f_2(\mathbf{x})$ is the computational parameter count (conversely reflecting the computational efficiency).

Non-dominated sorting and the calculation of crowding distance are integral parts of NSGA-II:

$$\text{NonDomSort}(\mathbf{X}) = \bigcup_{i=1}^{k} F_i \tag{4}$$

$$d(\mathbf{x}) = \sum_{i=1}^{k} (f_i(\mathbf{x}^+) - f_i(\mathbf{x}^-)) \tag{5}$$

Pareto front exploration and utilization

The Pareto front represents the set of non-dominated solutions, providing an optimal trade-off between conflicting objectives:

$$\mathcal{P} = \{\mathbf{x} \in \mathbf{X} \mid \nexists \mathbf{y} \in \mathbf{X} : \mathbf{F}(\mathbf{y}) \prec \mathbf{F}(\mathbf{x})\}, \tag{6}$$

where $\prec$ indicates that $\mathbf{y}$ dominates $\mathbf{x}$.

The Pareto front derived from the NAS process provides a practical framework for model selection:

- *Interpretation and analysis:* The front is analyzed to discern architectural trade-offs, allowing stakeholders to identify models with the desired balance between accuracy and efficiency.
- *Informed selection:* Decision-makers can select models that align with specific performance expectations and operational constraints by examining the Pareto front.
- *Resource allocation:* Models on the Pareto front inform resource allocation, directing investments toward architectures with the most favorable trade-offs.

Algorithm provides a detailed procedure that encapsulates our approach within the Optuna framework using NSGA-II [20] for the multi-objective optimization. This algorithm highlights, from initialization to final selection, the integration of the evolutionary search process with transformer-specific architecture and hyperparameter tuning.

Algorithm 1 Multi-Objective NAS for Time Series Anomaly Detection with TransNAS-TSAD

Input: Training dataset, validation dataset, predefined search space for transformer architectures and hyperparameters
Output: Optimized transformer architecture and hyperparameters for time series anomaly detection
Initialization: Set up the Optuna framework with the NSGA-II sampler. Initialize the search with a population of transformer architectures from the predefined search space. Define evolutionary process parameters: population size, number of generations, mutation and crossover rates.
Specify the multi-objective performance metric combining the accuracy (e.g., F1 score) and the computational efficiency (e.g., model parameter count).
for each study in Optuna **do**
  for each trial in the study **do**
    Define the transformer model architectural parameters and training hyperparameters within the search space constraints.
    Construct and train the transformer model on the training dataset.
    Evaluate the model on the validation dataset using the specified metrics.
    Update the study with the trial's objectives, reflecting the model's performance and computational efficiency.
    if early stopping criteria are met based on validation performance or predefined resource constraints **then**
      Terminate the trial.
    end if
  end for
  Selection: Use Optuna's NSGA-II sampler to select architectures for the next generation, focusing on the Pareto-optimal solutions.
  Crossover and Mutation: Apply Optuna's genetic operators to evolve the study's population, exploring new architectural variants.
end for
Final Selection: At the end of the NAS process, use Optuna to extract the best transformer architecture and hyperparameters from the Pareto front, considering both accuracy and computational efficiency.

3.3.3 Evolutionary process in TransNAS-TSAD

The evolutionary

process within TransNAS-TSAD is critical in achieving our overarching goal of refining anomaly detection capabilities. The process has three aspects, summarized below.

- *Generation loop*: New models are generated, evaluated, and passed on to each generation iteratively, with superior architectures refining the Pareto front.
- *Early stopping*: The process incorporates early stopping for models that do not show promise, thus conserving computational resources.
- *Final architecture selection*: The optimal architecture is selected from the Pareto front, ensuring an effective balance between the objectives.

Through iterative refinement and selection of model architectures, the Generation Loop aspect ensures that our models are not only accurate in identifying anomalies but also evolve to become more efficient and adaptable to various data characteristics. This continuous improvement and adaptation are key to achieving high-performance

anomaly detection, which stands at the core of our research objectives.

3.4 Offline optimization of transformer architecture for anomaly detection

Central to TransNAS-TSAD is the offline, strategic optimization of the transformer architecture, a process critical for achieving an effective balance between anomaly detection accuracy and computational efficiency. The optimization, guided by neural architecture search (NAS) using the NSGA-II algorithm, involves:

1. *Comprehensive trial-based exploration*: In line with Optuna's best practices, the process involves conducting over 100 trials to explore a wide array of transformer architectures, ensuring a thorough examination of the solution space.
2. *Achieving pareto front efficiency*: The key focus of NAS is to identify and refine a collection of architectures that constitute the Pareto front, representing a spectrum of high-performance, resource-intensive models to more balanced and resource-efficient alternatives.
3. *Iterative evolutionary process*: Through evolutionary algorithms, these architectures undergo continuous adaptation and improvement, exploring innovative configurations and enhancing accuracy and computational efficiency with each generation.

Tailoring architectures to specific deployment needs

After the offline NAS process, we are equipped with a diverse array of transformer architectures, each optimized for different operational contexts:

- *In resource-rich environments*: High accuracy (i.e., F1 score) models, though computationally demanding, are suitable for scenarios where resource is not a constrained.
- *In resource-limited environments*: The NAS process also yields architectures that are either balanced in terms of accuracy and efficiency or are specifically optimized for resource-limited environments.

This flexibility allows for the deployment of TransNAS-TSAD in various settings, ensuring efficient and effective anomaly detection tailored to the constraints and requirements of different deployment environments.

3.5 Advanced anomaly detection with adversarial elements in TransNAS-TSAD

TransNAS-TSAD represents a breakthrough in the field of time series anomaly detection by strategically enhancing adversarial learning paradigms. While it draws foundational inspiration from the TranAD model's two-phase approach, our framework introduces a neural architecture search (NAS) strategy that effectively incorporates a third, iterative, self-adversarial phase. This tripartite approach enables dynamic selection among the conventional two-phase mechanism and our advanced iterative phase, optimizing detection capabilities for each specific trial.

The inception of this third phase signifies a substantial evolution in anomaly detection techniques. By embedding adversarial elements within all three phases, TransNAS-TSAD not only retains the strengths of the traditional encoder-decoder models but also introduces a level of adaptability and precision previously unattainable. This method excels in identifying subtle and complex anomalies, leveraging the iterative adversarial training to refine detection with each iteration.

3.5.1 Three-phase adversarial approach

Drawing from adversarial learning paradigms, the three-phase approach in TransNAS-TSAD seeks to enhance the model's sensitivity to anomalies through competitive reconstruction stages.

Phase 1-preliminary input reconstruction

This phase, acting as the foundation, aims for a preliminary reconstruction of the input time-series window, and yields a focus score. The latter is defined by the deviation of the reconstructed output from the actual input:

$$L_{\text{focus}} = \|O_{\text{initial}} - W\|^2,$$

where O_{initial} is the output from the first phase and W is the input time-series window. The resultant focus score, derived from the deviations in this initial reconstruction, serves as an attention modulator for the subsequent phase.

Phase 2-adversarial, focus-driven reconstruction

Incorporating adversarial elements, this phase utilizes the focus scores from phase 1. In the adversarial phase, the second decoder aims to maximize the difference between its output and the input. Simultaneously, the first decoder aims to minimize this difference:

$$L_{\text{adv1}} = \|O_{\text{adv1}} - W\|^2$$

$$L_{\text{adv2}} = -\|O_{\text{adv2}} - W\|^2,$$

where O_{adv1} and O_{adv2} are the outputs from the first and second decoders, respectively, during the adversarial phase.

Phase 3-Iterative self-adversarial approach

Going beyond the structured two-phase reconstruction inspired by the TranAD work, TransNAS-TSAD introduces a dynamic iterative approach. This approach,

embedded with self-adversarial mechanisms, continually refines its reconstructions.

Iterative refinement:

Starting with an initial reconstruction, the deviation between the current output and the input provides feedback for the next iteration:

$$L_{\text{iteration}} = \|O_{\text{current}} - W\|_2$$

The self-adversarial mechanism can be represented as:

$$L_{\text{self-adv}} = (L_{\text{iteration,prev}} - L_{\text{iteration,current}})^2,$$

where $L_{\text{iteration,prev}}$ is the loss from the previous iteration and $L_{\text{iteration,current}}$ is the loss from the current iteration. The iterative refinement continues until the change in the loss between consecutive iterations falls below a predetermined threshold, signifying convergence. Mathematically, the convergence criterion can be defined as:

$$\Delta L = \left|L_{\text{iteration,current}} - L_{\text{iteration,prev}}\right|.$$

If $\Delta L < \epsilon$, where ϵ is a small positive value (e.g., 10^{-5}), the iteration stops, suggesting that further refinement may not yield significant improvements. After the iterative process converges, the best anomaly score is determined. The best score is derived from the iteration with the smallest reconstruction loss, ensuring that the selected representation most closely matches the input time-series data.

Adaptable and robust anomaly detection

What sets TransNAS-TSAD apart is its adaptability. The model is not confined to a fixed number of reconstruction phases. Depending on the intricacy of the dataset, it can dynamically adjust, ensuring that even the most subtle anomalies are not overlooked. Moreover, the incorporation of adversarial elements, both in the two-phase and iterative approaches, ensures the model continually challenges itself, leading to more refined and accurate reconstructions. TransNAS-TSAD represents a significant advancement in the realm of time-series anomaly detection. By amalgamating auto-regressive inference, adversarial mechanisms, attention modulation, and adaptive iterations, it offers a comprehensive solution, adept at detecting both overt and nuanced anomalies across varied time-series datasets.

3.5.2 Harnessing the power of TransNAS-TSAD for anomaly detection

Building upon the foundational architecture and adversarial mechanisms of TransNAS-TSAD, we now delve into the practical realm of anomaly detection. This section elucidates the methodologies employed to infer anomalies from time-series data, leveraging the trained transformer model and various augmentation strategies.

The anomaly scoring mechanism

Every incoming data point, denoted as $\hat{W}$, is assessed for anomalies by computing a score that quantitatively reflects its deviation from expected patterns. Here, $\hat{W}$ represents the new window of time series data under consideration. The anomaly score, s, is calculated as:

$$s = \frac{1}{2}\|R_1 - \hat{W}\|_2^2 + \frac{1}{2}\|\hat{R}_2 - \hat{W}\|_2^2, \tag{7}$$

where R_1 and $\hat{R}_2$ are the reconstructed outputs from the transformer model, corresponding to different stages of the inference process.

Dual inference pathways

TransNAS-TSAD uses two distinct inference methodologies:

- *Two-phase approach*: A structured two-step inference process yielding reconstructions $(R_1, \hat{R}_2)$.
- *Iterative refinement*: For scenarios demanding intricate attention, it engages in iterative inference, perfecting the anomaly scores with each iteration.

Figure 2 illustrates the anomaly detection process on dimension 0 of the Server Machine dataset (SMD) test dataset using TransNAS-TSAD. The upper plot displays the true data points (in blue) and the predicted values (in red), while the lower plot indicates the computed anomaly scores and their corresponding labels, offering a visual interpretation of the model's detection capabilities in identifying anomalies.

Dynamic anomaly thresholding: the sentinel

Central to our thresholding strategy is the peaks over threshold (POT) method. At its core, POT establishes a threshold where observations surpassing this mark are deemed anomalous, allowing for effective discernment of extreme data points.

Evolutionary in nature, our modified POT (mPOT) approach ensures adaptability to the ever-changing landscape of time-series data:

$$mPOT(x) = POT(x) + \alpha \times \text{recent\_deviation}(x), \tag{8}$$

where α is a weight. The function recent_deviation calculates the deviation of the latest data points from their median value. Anomalies are pinpointed whenever any dimension's score, s_i, exceeds this dynamic mPOT threshold.

3.5.3 Augmentative strategies for enhanced precision

TransNAS-TSAD employs various augmentative strategies to bolster the precision of the POT method, ensuring that the anomaly detection mechanism remains sensitive, adaptable, and robust. These strategies aim to enhance the detection capabilities by refining the anomaly scores and the thresholds against which they are evaluated.

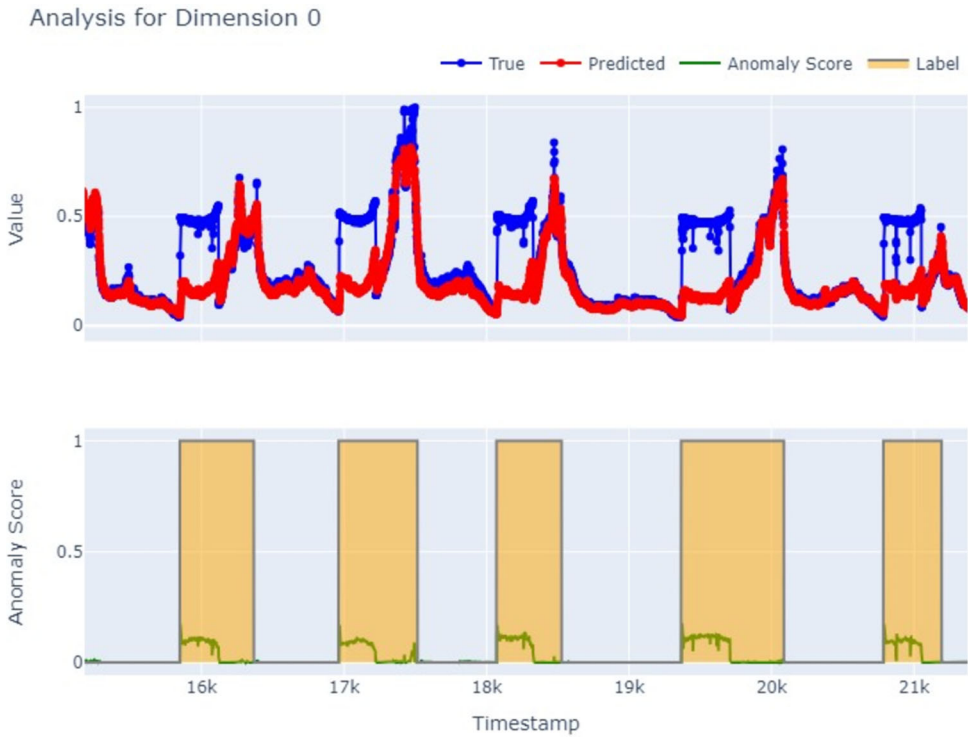

Fig. 2 Anomaly detection in TransNAS-TSAD on dimension 0 of the SMD test dataset, depicting true versus predicted values and identified anomalies

Moving average thresholding (MAT)

MAT is a dynamic thresholding technique that complements the POT method. Instead of using a static threshold, MAT calculates a moving average of recent anomaly scores to adaptively set the threshold. This dynamic adjustment ensures that the threshold is responsive to emerging data trends and patterns, enhancing its relevance and accuracy. The moving average threshold at time t is given by:

$$MAT(t) = \frac{1}{N} \sum_{i=t-N}^{t} s_i, \tag{9}$$

where s_i is the anomaly score at time i and N represents the window size for the moving average.

Rolling statistics for nuanced detection

Rolling statistics, particularly the rolling mean and standard deviation, provide additional context to the POT method by capturing temporal dependencies and trends in the data. These statistics are instrumental in uncovering subtle anomalies that might otherwise be overlooked. For each data point at time t, the rolling mean $\mu(t)$ and rolling standard deviation $\sigma(t)$ are computed as:

$$\mu(t) = \frac{1}{W} \sum_{i=t-W}^{t} x_i \quad \text{and} \quad \sigma(t) = \sqrt{\frac{1}{W} \sum_{i=t-W}^{t} (x_i - \mu(t))^2}, \tag{10}$$

where x_i represents the data point at time i and W is the rolling window size. These rolling statistics are incorporated into the feature set, enhancing the model's ability to discern intricate data variations and improving the relevance of the anomaly scores generated for the POT method.

4 Experimental setup for TransNAS-TSAD

We evaluate the performance of TransNAS-TSAD against a suite of established benchmark models, including TranAD [16], LSTM-NDT [50], DAGMM [51], OmniAnomaly [52], MSCRED [53], MAD-GAN [54], USAD [55], CAE-M [56]), and GDN [57]. To ensure a fair comparison, we utilize the hyperparameter configurations as specified in the original publications of these models, relying on their publicly available implementations.

The experimental infrastructure comprises a Google Colab Pro environment, leveraging a NVIDIA Tesla T4 GPU (16GB memory) and an Intel Xeon E5-2670 v3 CPU (8 cores, 51GB memory). The NSGA-II-based NAS algorithm and data management operations are implemented in Python, utilizing robust libraries such as PyTorch for deep learning, Optuna for hyperparameter optimization, and pandas for data handling.

A pivotal element of our methodology is the neural architecture search (NAS), which automates the architectural design by navigating a comprehensive search space of architectural and training hyperparameters, shown in Table 1. This search space is crafted to enable the NAS

algorithm to adapt the model architecture to the unique characteristics of each dataset. Consistent with the practices in OmniAnomaly and TranAD, we apply an enhanced version of the peaks over threshold (POT) method [9] across all datasets with a uniform coefficient of 10^{-4}. This enhancement incorporates our previously described augmentative strategies, further fine-tuning the low quantile parameter for each specific dataset to align with established benchmarks and ensure equitable comparative analysis.

4.1 Datasets

Our study employs a suite of datasets selected to challenge and validate the robustness of the time series anomaly detection methods across a variety of real-world scenarios. These datasets, renowned for their complexity and diversity, allow for an effective benchmarking of our TransNAS-TSAD approach against existing methodologies. See Table 2 for summary statistics.

- *Numenta anomaly benchmark (NAB) dataset*: This dataset encompasses a wide range of real-world data, including temperature sensor readings, cloud machine CPU utilization, service request latencies, and taxi demand data in New York City. It is important to note some anomalies in labeling within this dataset are excluded from analyses, particularly in the NYC-taxi traces [58, 59].
- *HexagonML (UCR) dataset*: Featured in the KDD 2021 cup, this dataset comprises a diverse collection of univariate time series. In our research, we selectively use natural data representations derived from real-world sources, focusing specifically on the InternalBleeding and ECG datasets. Consequently, we did not include any synthetic sequences that are also part of the HexagonML (UCR) dataset [60].
- *MIT-BIH supraventricular arrhythmia database (MBA) dataset*: This dataset includes electrocardiogram recordings from four patients, featuring anomalies such as supraventricular contractions or premature heartbeats. It's a well-recognized dataset in data management studies [61, 62].
- *Soil moisture active passive (SMAP) dataset*: This NASA-provided dataset includes global measurements of soil moisture in the top 5 cm of Earth's soil surface, collected approximately every three days by the SMAP satellite. It is designed to enhance our understanding of water, carbon, and energy cycles[50].
- *Mars science laboratory (MSL) dataset*: Similar to the SMAP dataset, the MSL dataset includes sensor and actuator data from the Mars rover.[50] Due to the presence of many trivial sequences, only specific non-trivial sequences are typically analyzed such as (A4, C2 and T1) pointed by [16, 59].
- *Secure water treatment (SWaT) dataset*: This dataset is derived from a real-world water treatment plant, including data from seven days of normal operations and four days under abnormal conditions. It features readings from various sensors and actuators [63].
- *Water distribution (WADI) dataset*: An expansion of the SWaT dataset, WADI includes a larger array of sensors and actuators. The dataset spans a longer period, with 14 days of normal operation and two days under attack scenarios [64].
- *Server machine dataset (SMD)*: Covering five weeks of data, this dataset includes resource utilization traces from 28 machines in a compute cluster. Only specific non-trivial sequences are used for analysis [52].

4.2 Evaluation criteria

We employ key metrics aligned with the objectives of the TransNAS-TSAD framework. These metrics are selected to optimize anomaly detection effectiveness, considering practical deployment aspects. The F1 score, defined as the harmonic mean of precision and recall, is central to our evaluation strategy. It is an accuracy measure that balances the trade-off between false positives and false negatives to help address the challenge of imbalanced datasets common in anomaly detection.

Table 2 Dataset statistics

Dataset	Train size	Test size	Dimensions	Sequences	Anomalies (%)
NAB	4033	4033	1	6	0.92
UCR	1600	5900	1	4	1.88
MBA	100,000	100,000	2	8	0.14
SMAP	135,183	427,617	25	55	13.13
MSL	58,317	73,729	55	3	10.72
SWaT	496,800	449,919	51	1	11.98
WADI	1,048,571	172,801	123	1	5.99
SMD	708,405	708,420	38	4	4.16

4.2.1 Introducing the efficiency-accuracy-complexity score (EACS)

Model complexity is frequently measured by the parameter count, a metric that, while useful, does not fully capture the capabilities of transformer-based models like TransNAS-TSAD. Despite their high parameter count, these models are highly effective at capturing complex data patterns, a strength that justifies their complexity. Their advanced processing capabilities, facilitated by parallelization, make them adept at handling scenarios that demand high accuracy, even when computational resources are abundant. Therefore, when we report the parameter count of TransNAS-TSAD, it is within the broader context of its sophisticated functionality. Additionally, our evaluation emphasizes not only the parameter count but also the F1 score and Efficiency-Accuracy-Complexity Score (EACS), ensuring that our assessment considers both the model's anomaly detection effectiveness and its deployability. While other metrics like ROC/AUC are valuable, our focused approach aligns with the specific objectives of TransNAS-TSAD, aiming for a balanced evaluation of its performance.

The weights for the Efficiency-Accuracy-Complexity Score (EACS) are set based on the relative importance of each factor-accuracy, training efficiency, and model complexity-in real-world deployment scenarios. Typically, the weights are chosen from a range of 0 to 1, with the sum of weights equal to 1. For instance, in applications where accuracy is of paramount importance, the accuracy weight (w_a) is set higher, often between 0.4 to 0.6, while the weights for training time (w_t) and model complexity (w_p) are slightly lower, ranging from 0.2 to 0.4. This weighting scheme is flexible, allowing it to be adjusted based on specific deployment needs, balancing performance and resource constraints. In this study, we set the weights as follows: $w_a = 0.4$, $w_t = 0.4$, and $w_p = 0.2$, reflecting the need for high accuracy while maintaining practical efficiency and manageable model complexity for deployment.

To conduct a fair and comparative assessment across models, we have calculated the number of parameters, training time, and F1 score for each benchmark model as well as for the best model instance obtained from TransNAS-TSAD for each dataset. The maximum values for F1 score, training time, and parameter count-represented as $\mathrm{F1}_{\max}, \mathrm{T}_{\max}, \mathrm{P}_{\max}$-are the highest values observed among all models compared for each specific dataset. This normalization ensures a fair comparison across models. The EACS is defined as:

$$\mathrm{EACS} = w_a \times \left(\frac{\mathrm{F1}}{\mathrm{F1}_{\max}}\right) + w_t \times \left(1 - \frac{\mathrm{T}_{\mathrm{train}}}{\mathrm{T}_{\max}}\right) + w_p \times \left(1 - \frac{\mathrm{P}_{\mathrm{count}}}{\mathrm{P}_{\max}}\right) \tag{11}$$

where F1 is the model accuracy given as the F1 score, $\mathrm{T}_{\mathrm{train}}$ is the training time, and $\mathrm{P}_{\mathrm{count}}$ is the parameter count of the model which indicates the complexity of the model, and w_a, w_t, and w_p are, respectively, the weights of the corresponding performance factors. These weights were chosen to reflect the importance of accuracy and training efficiency in practical deployments, which are often prioritized over the complexity of the model.

5 Results

Evaluating TransNAS-TSAD: precision, recall, and F1 score analysis

The evaluation of anomaly detection models, as shown in Table 3, measures TransNAS-TSAD's performance across diverse datasets including NAB, UCR, MBA, SMAP, MSL, SWaT, WADI, and SMD, using precision, recall, and F1 scores as benchmarks. Bold values in the tables indicate the best performance for each dataset, highlighting superior F1 scores. TransNAS-TSAD demonstrated high F1 scores in datasets such as NAB, UCR, MBA, and SMAP, benefiting from its advanced data processing techniques that adeptly handle complex patterns in both univariate and multivariate time series. This distinguishes it from models like OmniAnomaly and MSCRED. For the MSL dataset, the TranAD model slightly outperforms TransNAS-TSAD (0.9494 vs. 0.9482 in the F1 score), likely due to TranAD's parameters being finely tuned to MSL's unique characteristics. Despite conducting over 100 trials per dataset, this suggests that even more exhaustive optimization could potentially unlock further improvements for TransNAS-TSAD. Its robust performance in the SWaT dataset, with an F1 score of 0.8314, underscores its versatility in various industrial contexts. In the cases of WADI and SMD, where TransNAS-TSAD achieves F1 scores of 0.8400 and 0.9986, respectively, the significant improvements-such as a 40% increase over baseline in WADI-are attributable to its comprehensive search space and optimization strategies, enabling effective model tuning. This evaluation underscores the potential benefits of extending our optimization framework to achieve even greater model refinement.

The variable performance of models like LSTM-NDT and DAGMM across different datasets points to their methodological strengths and constraints. For example, the LSTM-NDT's reduced efficacy likely stems from the method's nonparametric thresholding and resulting sensitivity to large differences in anomaly patterns; and the DAGMM struggles with longer sequences, in part, due to its reliance on a singular GRU model, which limits its

Table 3 Benchmarking TransNAS-TSAD against the baseline models: A summary of precision (P), recall (R), and F1 scores showcases the advanced anomaly detection capabilities of TransNAS-TSAD, derived via NSGA-II-based NAS, across diverse datasets

Method	NAB			UCR			MBA			SMAP		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LSTM-NDT	0.6400	0.6667	0.6531	0.5231	0.8294	0.5231	0.9207	0.9718	0.9456	0.8523	0.7326	0.7879
DAGMM	0.7622	0.7292	0.7453	0.5337	0.9718	0.5337	0.9475	0.9900	0.9683	0.8069	0.9891	0.8888
OmniAnomaly	0.8421	0.6667	0.7442	0.8346	0.9999	0.8346	0.8561	1.000	0.9225	0.8130	0.9419	0.8728
MSCRED	0.8522	0.6700	0.7502	0.5441	0.9718	0.5441	0.9272	1.0000	0.9623	0.8175	0.9216	0.8664
MAD-GAN	0.8666	0.7012	0.7752	0.8538	0.9891	0.8538	0.9396	1.0000	0.9689	0.8157	0.9216	0.8654
USAD	0.8421	0.6667	0.7442	0.8952	1.0000	0.8952	0.8953	0.9989	0.9443	0.7480	0.9627	0.8419
CAE-M	0.7918	0.8019	0.7968	0.6981	1.0000	0.6981	0.8442	0.9997	0.9154	0.8193	0.9567	0.8827
GDN	0.8129	0.7872	0.7998	0.6894	0.9988	0.6894	0.8832	0.9892	0.9332	0.7480	0.9891	0.8518
TranAD	0.8889	0.9892	0.9364	0.9407	1.0000	0.9407	0.9569	1.0000	0.9780	0.8043	0.9999	0.8915
TransNAS-TSAD	0.8888	1.0000	**0.9411**	0.9823	1.0000	**0.9910**	0.9726	1.0000	**0.9861**	0.9066	1.0000	**0.9510**

Method	MSL			SWaT			WADI			SMD		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LSTM-NDT	0.6288	1.0000	0.7721	0.7778	0.5109	0.6167	0.0138	0.7823	0.0271	0.9736	0.8440	0.9042
DAGMM	0.7363	1.0000	0.8482	0.9933	0.6879	0.8128	0.0760	0.9981	0.1412	0.9103	0.9914	0.9491
OmniAnomaly	0.7848	0.9924	0.8765	0.9782	0.6957	0.8131	0.3158	0.6541	0.426	0.8881	0.9985	0.9401
MSCRED	0.8912	0.9862	0.93693	0.9992	0.6770	0.8072	0.2513	0.7319	0.3741	0.7276	0.9974	0.8414
MAD-GAN	0.8516	0.9930	0.9169	0.9593	0.6957	0.8065	0.2233	0.9124	0.3588	0.9991	0.8440	0.915
USAD	0.7949	0.9912	0.8822	0.9977	0.6879	0.8143	0.1873	0.8296	0.3056	0.9060	0.9974	0.9495
CAE-M	0.7751	1.0000	0.8733	0.9697	0.6957	0.8101	0.2782	0.7918	0.4117	0.9082	0.9671	0.9367
GDN	0.9308	0.9892	**0.9591**	0.9697	0.6957	0.8101	0.2912	0.7931	0.4260	0.7170	0.9974	0.8342
TranAD	0.9038	0.9999	0.9494	0.9760	0.6997	0.8151	0.3529	0.8296	0.4951	0.9262	0.9974	0.9605
TransNAS-TSAD	0.9567	1.000	0.9482	0.9415	0.7624	**0.8314**	0.8508	0.8295	**0.8400**	0.9985	0.9988	**0.9986**

capacity to accurately map extended temporal dynamics. Moreover, TransNAS-TSAD demonstrates enhanced performance over innovative yet less adaptable models such as MAD-GAN and USAD, and while CAE-M and GDN show potential in specific scenarios, they generally fall short of matching TransNAS-TSAD's applicability and performance across the datasets reviewed, highlighting the value of integrating adversarial training and iterative optimization into the TransNAS-TSAD's framework.

In-depth analysis of training efficiency and EACS in anomaly detection models

Table 4 presents a comprehensive analysis that compares the Efficiency-Accuracy-Complexity Score (EACS) among various anomaly detection models, including TransNAS-TSAD. As discussed earlier in Section 4.2.1, EACS is a composite score that balances accuracy (F1 score), training efficiency, and model complexity to assess the overall practicality of a model in real-world deployments. This balance is critical, especially in operational environments that require rapid deployment and frequent updates, as highlighted by several studies [65–67]. In this analysis, TransNAS-TSAD emerges as a standout performer, achieving high EACS values across multiple datasets, indicating its ability to maintain strong model performance while ensuring operational efficiency.

Specifically, TransNAS-TSAD delivers high F1 scores of 94.11% and 99.10% on the NAB and UCR datasets, respectively, while maintaining brief training periods of 2.70 and 2.24 s. This translates into EACS values of 0.9742 and 0.9922, showcasing TransNAS-TSAD's ability to adapt quickly to complex data while balancing accuracy and efficiency. The trend—high accuracies combined with short training times—remains consistent across other datasets, including MBA and SMAP, where TransNAS-TSAD records EACS scores of 0.9932 and 0.9734, respectively.

EACS helps to provide insight into trade-offs between performance and resource consumption. For example, models like MSCRED, despite achieving commendable F1 scores, experience a notable decrease in EACS due to

Table 4 Comparative analysis of the Efficiency-Accuracy-Complexity Score (EACS) across various anomaly detection models, highlighting the performance of TransNAS-TSAD. The EACS is calculated by normalizing and weighting F1 scores (0.4), training times (0.4), and parameter counts (0.2), demonstrating the balanced efficiency and accuracy of TransNAS-TSAD against benchmark models across multiple datasets

Method	NAB				UCR			
	F1 %	Training time (sec)	# Params	EACS	F1 %	Training Time (sec)	# Params	EACS
LSTM-NDT	65.31	23.40	1710	0.8424	52.31	11.14	1765	0.7985
DAGMM	74.53	64.50	1266	0.8467	53.37	27.50	1214	0.7919
OmniAnomaly	74.42	88.70	13,717	0.8250	83.46	38.21	13,025	0.8863
MSCRED	75.02	503.60	1,237,377	0.3001	54.41	559.60	128,655	0.2176
MAD-GAN	77.52	53.90	838	0.8671	85.38	31.50	822	0.9177
USAD	74.42	65.40	1359	0.8455	89.52	54.23	1359	0.9172
CAE-M	79.68	33.60	7229	0.8909	69.81	22.10	7365	0.8520
GDN	79.98	131.60	574	0.8153	68.94	64.70	**566**	0.8286
TranAD	93.64	3.37	615	0.9718	94.07	**1.14**	619	0.9745
TransNAS-TSAD	**94.11**	**2.70**	**285**	**0.9742**	**99.10**	2.24	1690	**0.9922**

Method	MBA				SMAP			
	F1 %	Training time (sec)	# Params	EACS	F1 %	Training Time (sec)	# Params	EACS
LSTM-NDT	94.56	47.80	18,641	0.9520	78.79	36.43	29,146	0.8459
DAGMM	96.83	92.36	2448	0.9394	88.80	26.60	7266	0.9050
OmniAnomaly	92.25	197.66	25,474	0.8649	87.28	66.77	16,813	0.8230
MSCRED	96.23	774.99	2,441,778	0.3849	86.64	25.70	8,237,452	0.6982
MAD-GAN	96.89	229.78	1877	0.8688	86.54	41.26	6718	0.8683
USAD	94.43	191.26	1609	0.8789	84.19	49.22	7395	0.8439
CAE-M	91.54	89.66	15,411	0.9186	88.27	212.51	7229	0.5529
GDN	93.32	203.45	1106	0.8682	85.18	104.50	**2974**	0.7440
TranAD	97.80	5.11	1298	0.9885	89.15	5.62	62,271	0.9445
TransNAS-TSAD	**98.61**	**2.27**	**593**	**0.9932**	**94.60**	**3.19**	39,045	**0.9734**

Method	MSL				SWaT			
	F1 %	Training time (sec)	# Params	EACS	F1 %	Training Time (sec)	# Params	EACS
LSTM-NDT	77.21	37.80	61,856	0.8887	66.70	41.50	72,854	0.7957
DAGMM	84.82	25.66	18,756	0.9258	81.28	29.66	16,558	0.8748
OmniAnomaly	87.65	29.69	39,753	0.9348	81.31	40.20	36,541	0.8568
MSCRED	93.63	40.70	18,476,728	0.7535	80.72	236.80	14,227,264	0.3229
MAD-GAN	91.69	32.64	17,446	0.9497	80.65	34.70	15,958	0.8638
USAD	88.22	29.60	**14,859**	0.9374	81.43	35.81	**12,502**	0.8651
CAE-M	87.33	774.60	204,687	0.5471	81.01	71.22	194,525	0.8010
GDN	**95.91**	121.80	153,541	0.9191	81.01	83.55	115,442	0.7813
TranAD	94.94	6.88	272,181	0.9733	81.53	2.10	31,336	0.9221
TransNAS-TSAD	94.82	**5.44**	236,491	**0.9739**	**83.14**	**1.60**	13,464	**0.9297**

Method	WADI				SMD			
	F1 %	Training time (sec)	# Params	EACS	F1 %	Training Time (sec)	# Params	EACS
LSTM-NDT	2.71	422.67	133,712	0.5753	90.42	460.70	42,853	0.9100
DAGMM	14.12	246.32	39,387	0.6498	94.91	337.90	10,516	0.9420
OmniAnomaly	42.60	361.60	39,753	0.7794	94.01	311.20	22,941	0.9412

Table 4 (continued)

Method	WADI				SMD			
	F1 %	Training time (sec)	# Params	EACS	F1 %	Training Time (sec)	# Params	EACS
MSCRED	37.41	1884.25	75,530	0.6716	84.14	349.77	14,237,356	0.6978
MAD-GAN	35.88	321.20	35,764	0.7500	91.50	424.60	9903	0.9188
USAD	30.56	389.72	**32,859**	0.7214	94.95	328.77	10,609	0.9432
CAE-M	41.17	7724.92	388,905	0.3503	93.67	3606.90	114,219	0.5731
GDN	42.60	6047.12	287,082	0.4559	83.42	1000.40	**7255**	0.8226
TranAD	49.51	**177.60**	1,378,173	0.6645	96.05	56.70	135,110	0.9760
TransNAS-TSAD	**84.35**	227.50	1,701,515	**0.7883**	**99.81**	**52.40**	132,050	**0.9916**

prolonged training times (e.g., 774.99 s for MBA). This contrast emphasizes the advantage of TransNAS-TSAD, which achieves superior efficiency by optimizing both model complexity and training time while maintaining competitive F1 scores.

Moreover, EACS emphasizes the importance of practical deployability, which is particularly evident in datasets with challenging environments like SWaT and WADI. Here, TransNAS-TSAD not only sustains high F1 scores but also significantly reduces training times, achieving EACS scores of 0.9297 and 0.7883, respectively. This balance makes TransNAS-TSAD more suitable for environments where both accuracy and efficiency are required. While models like GDN and TranAD perform well in specific datasets, they do not consistently match TransNAS-TSAD in training efficiency across the board. EACS serves as an important metric in this evaluation, as it highlights cases where models with high accuracy may fall short due to resource demands, providing a more comprehensive evaluation of their practical applicability.. This underscores the value of considering training time, model size, and accuracy together when deploying anomaly detection models in real-world scenarios.

5.1 Optimal model configurations identified by TransNAS-TSAD

Table 5 showcases the culmination of TransNAS-TSAD's NSGA-II optimization process, highlighting the training and architectural hyperparameters of models that achieved the best balance between high F1 scores and low parameter counts. These configurations represent models uniquely suited to each dataset's anomaly detection needs.

A key observation is the diversity in architectural parameters across datasets, reflecting TransNAS-TSAD's adeptness in tailoring models to specific data characteristics. For example, in the SMAP dataset, the model employs a relatively simple architecture with fewer encoder and decoder layers, which is effective in handling the environmental time series data of this dataset. In contrast, the WADI dataset, known for its complex sensor network, necessitates a more intricate model structure, evident in its higher number of attention heads and the use of a two-phase reconstruction and refinement strategy.

Similarly, the variation in window sizes, ranging from 12 data points for SMD to 30 data points for WADI, underscores the model's flexibility in adapting to the temporal scale of different datasets. Larger window sizes in datasets like WADI allow the model to capture longer-term dependencies and subtle anomalies in extensive time series data, a crucial requirement for sophisticated water-related infrastructures.

The adaptability of TransNAS-TSAD is further demonstrated in its choice of positional encoding and layer normalization techniques, which vary significantly among datasets. For instance, the sinusoidal positional encoding in SMAP and NAB caters to their unique temporal patterns, while the Fourier positional encoding in UCR and WADI aligns with the datasets' complex spectral characteristics.

These results from TransNAS-TSAD's NAS process validate the framework's ability to not only fine-tune hyperparameters for operational efficiency, but also to intricately adapt its architectural design to the nuanced demands of various time series anomaly detection scenarios, ensuring optimal detection performance across a broad spectrum of datasets.

5.2 Architectural and training hyperparameter importance for F1 score optimization

The optimization of F1 scores in the TransNAS-TSAD model shown in Fig. 3 is critically analyzed through hyperparameter importance plots for four datasets: NAB, MBA, SMAP, and WADI. These plots offer quantitative insights into the relative impact of various hyperparameters

Table 5 Hyperparameter optimization results: Detailed summary of the optimal training and architectural hyperparameters identified for each dataset, demonstrating the adaptive precision of TransNAS-TSAD in time series anomaly detection

Arch and training params	SMAP	UCR	MBA	SWaT	MSL	SMD	NAB	WADI
Learning rate	0.0002128	0.0019997	0.003441	0.0000403	0.006192	0.0002273	0.006924	0.0015406
Dropout Rate	0.2353	0.4474	0.1795	0.3836	0.3730	0.1554	0.4560	0.2886
Dim Feedforward	101	124	41	42	121	86	10	24
Batch Size	32	48	16	32	32	48	96	128
Encoder Layers	2	1	3	2	2	3	1	2
Decoder Layers	1	2	1	3	3	1	2	1
Activation Func	sigmoid	leaky_relu	tanh	tanh	leaky_relu	sigmoid	relu	leaky_relu
Time Warping	False	True	True	True	True	False	False	True
Time Masking	True	True	False	True	True	False	True	False
Gaussian Noise	0.000151	0.027956	0.007925	0.058019	0.000428	0.000628	0.000119	0.001042
Linear Embedding	False	True	False	False	True	True	True	True
Phase Type	2phase	iterative	iterative	2phase	2phase	2phase	iterative	2Phase
Self-Conditioning	False	True	False	True	True	False	True	False
Layer Norm	False	False	False	True	True	True	False	False
Pos. Enc. Type	sinusoidal	fourier	fourier	sinusoidal	fourier	sinusoidal	sinusoidal	sinusoidal
FFN Layers	1	1	1	3	2	3	1	1
Attn Heads	25	1	2	51	55	38	1	127
Window Size	10	20	14	22	26	12	18	26

on the F1 score, aiding the fine-tuning process in our NAS-driven anomaly detection model.

For the NAB dataset, `gaussian_noise` and `activation_function` are the most influential hyperparameters, with importances of 0.18 and 0.15, respectively. The incorporation of `gaussian_noise` enhances the model's generalization capabilities, essential for robustness, while the choice of `activation_function` is key for capturing the data's nonlinear relationships.

The MBA dataset's analysis underscores the importance of `activation_function` as 0.11, `num_feedforward_layers` as 0.07, and `dropout_rate` as 0.07, indicating the need for a complex model architecture to effectively learn from multivariate ECG time series.

In the SMAP dataset, `gaussian_noise` with importance 0.53 highlights the model's ability to handle noisy environmental data effectively. Additionally, `batch_size` with importance 0.07 influences the model's performance, reflecting the impact of batch processing on training.

The WADI dataset, with its extensive sensor network, prioritizes `activation_function` with importance 0.16 and `dropout_rate` with importance 0.08 to maintain model robustness and prevent overfitting in high-dimensional spaces.

The recurrence of `activation_function` and `dropout_rate` across datasets emphasizes their role in nonlinear data transformation and regularization. This pattern reflects the impact of data quality and structure on the model's learning efficacy.

This hyperparameter analysis illuminates the tailored performance of TransNAS-TSAD across varied datasets, with each dataset's specifics dictating the importance of particular hyperparameters. This not only affirms the model's adaptability but also its capacity for ongoing enhancement, ensuring it remains adept at confronting the evolving complexities of time series data.

5.3 Pareto front analysis for model optimization

Building upon the insights from the hyperparameter importance analysis, the Pareto front optimization results for the TransNAS-TSAD model are depicted in Fig. 4. The Pareto front plots for the NAB, MBA, SMAP, and WADI datasets underscore the delicate balance between model complexity, as measured by the number of parameters, and the model's effectiveness, as quantified by the F1 score.

For the NAB dataset, the Pareto front indicates a dense congregation of model configurations. A significant number of these configurations exhibit a high parameter count without a corresponding increase in the F1 score, suggesting a potential plateau in performance gains relative to complexity. This observation prompts a critical evaluation of model parsimony, encouraging the selection of simpler

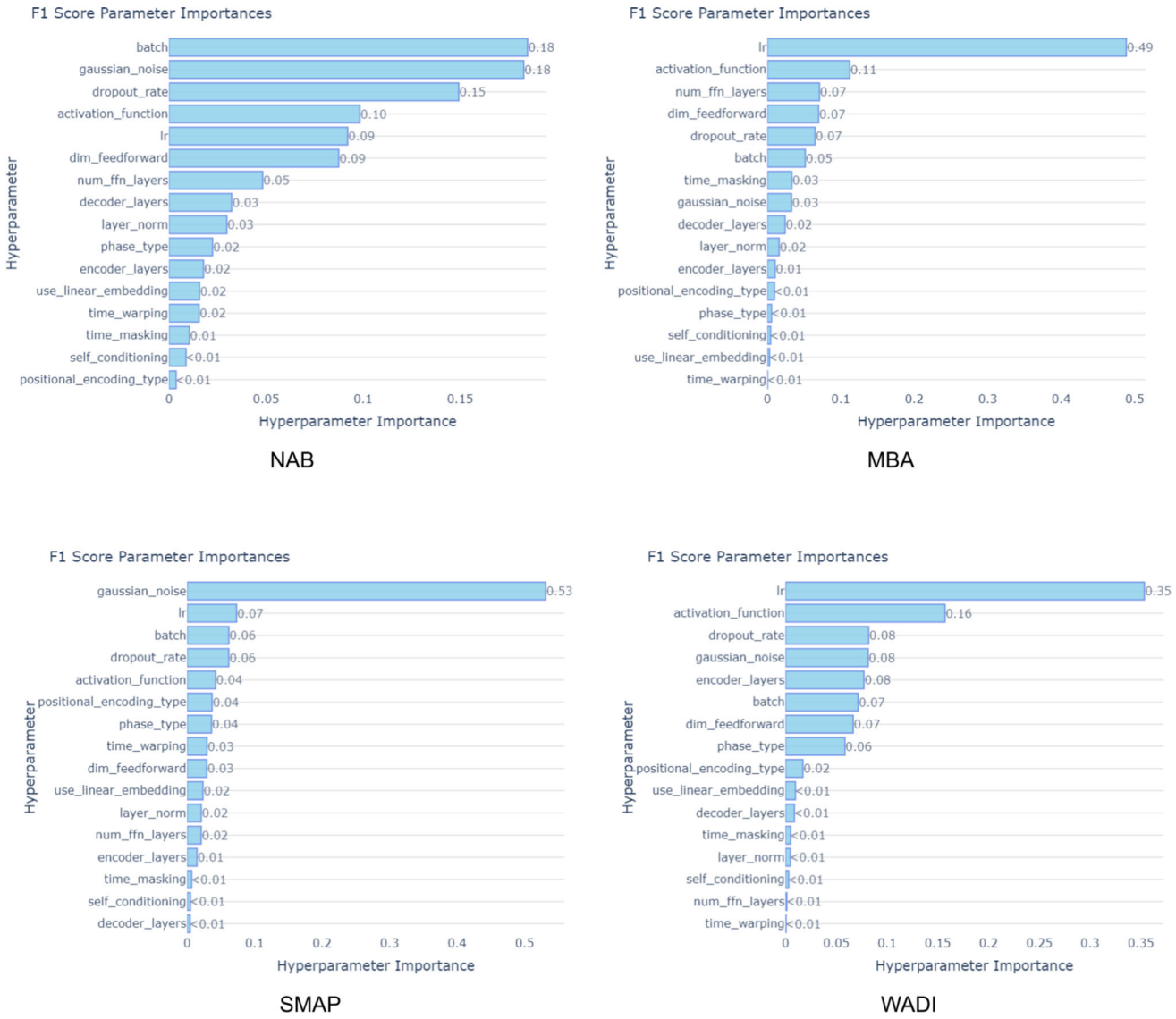

Fig. 3 Analysis of hyperparameter importance for F1 score optimization in TransNAS-TSAD across four datasets (NAB, MBA, SMAP, and WADI). The plots illustrate the relative impact of different hyperparameters on the F1 score, guiding the model's fine-tuning process for effective anomaly detection in diverse time series datasets

models that maintain performance while mitigating the risk of over-fitting.

In the MBA dataset, the dispersion of trials across the Pareto front reflects a comprehensive exploration of the architectural space. Interestingly, several models achieve commendable F1 scores without a proportionate surge in parameters, highlighting efficient architectural choices that capture the essential dynamics of ECG time series data without unnecessary complexity.

The SMAP dataset presents an outlier model with a substantial number of parameters yet achieving a high F1 score. This result may point to an overfitting scenario where the model's complexity does not translate into generalized performance. Such an insight is invaluable for guiding the model selection process toward architectures that balance accuracy with generalization.

The Pareto front for the WADI dataset, characterized by its complex sensor network, illustrates the necessity of sophisticated models to enhance the F1 score. The trend of increasing model complexity to achieve incremental improvements in performance is evident, underlining the inherent challenges of anomaly detection in high-dimensional industrial control systems.

These Pareto fronts provide a visual and quantitative tool for identifying optimal model configurations that achieve a balance between accuracy and complexity (as the converse of efficiency). They serve as a decision-making

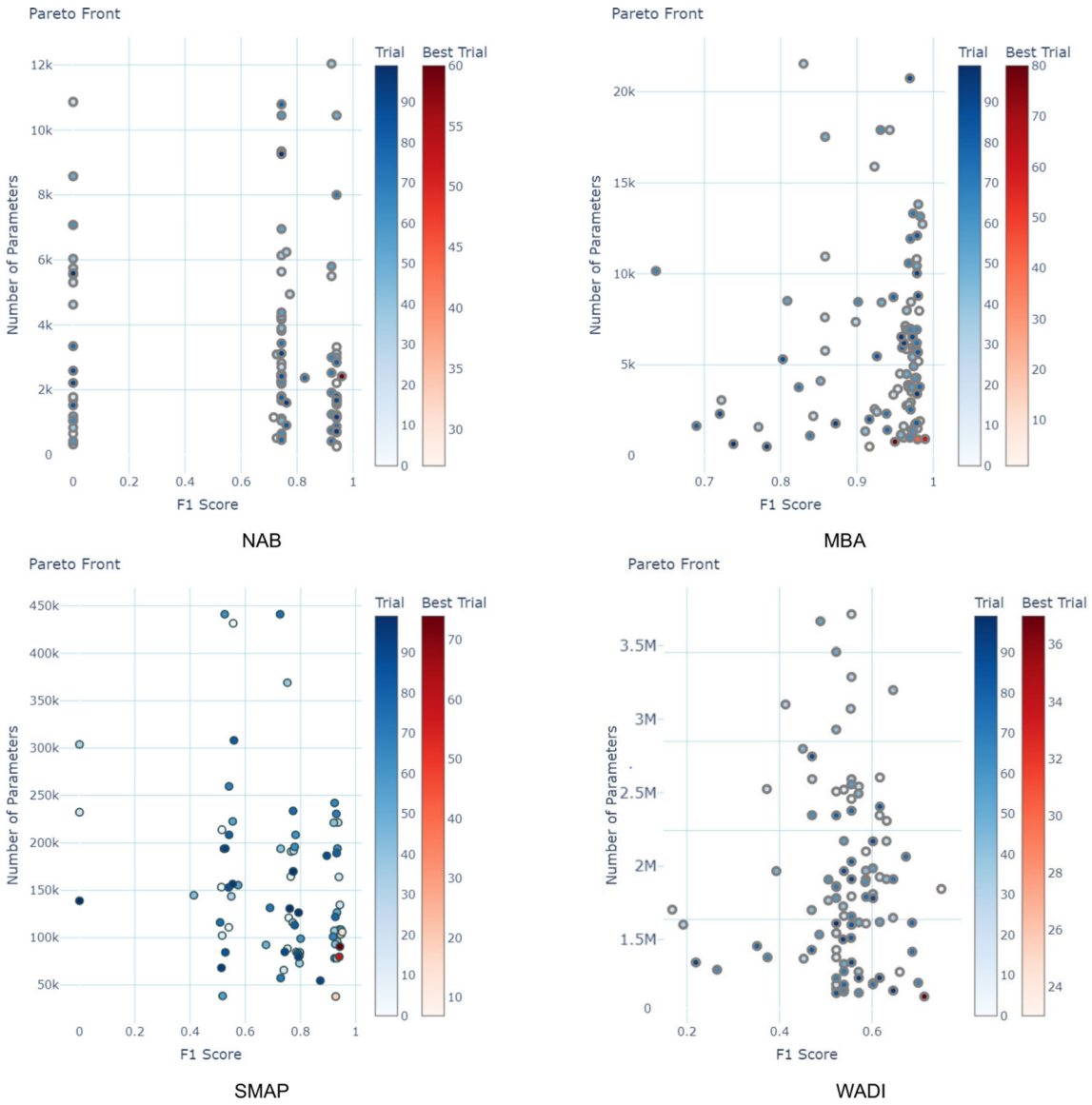

Fig. 4 Pareto front plots illustrating the trade-off between F1 score and number of parameters for NAB, MBA, SMAP, and WADI datasets in the TransNAS-TSAD optimization process

aid for selecting models that align with the practical demands of anomaly detection in diverse environments.

The findings from the Pareto analysis are integral to the ongoing development of the TransNAS-TSAD model within the broader scope of our research. They contribute to understanding how different model architectures perform across various datasets, and offer a pathway to enhancing the model's adaptability and ensuring its continued efficacy in the face of evolving data challenges.

6 Discussion: challenges and future directions

TransNAS-TSAD represents a significant advancement in anomaly detection within time series data, primarily through its innovative integration of advanced adversarial learning paradigms, NSGA-II optimization, and transformer architecture optimization. This confluence of technologies marks a substantial leap forward, particularly in the detection of subtle and complex anomalies that have typically eluded traditional methods.

A pivotal challenge addressed by TransNAS-TSAD is the balance between detection accuracy and computational efficiency. The NSGA-II algorithm plays a critical role in

optimizing model performance, without incurring excessive computational demands. This aspect is particularly crucial in real-world applications where resources are finite and efficiency is paramount.

The adaptability of TransNAS-TSAD is further exemplified in its dynamic adjustment capabilities, allowing it to effectively respond to the unique characteristics of different datasets. This adaptability is essential in the domain of anomaly detection, where dataset variability can present diverse challenges. Additionally, the iterative self-adversarial approach employed by TransNAS-TSAD significantly enhances detection accuracy, showcasing the model's sophisticated capabilities in identifying anomalies.

However, challenges remain, particularly in the realm of ensuring model generalization across a diverse array of datasets. The Pareto front analysis within TransNAS-TSAD has highlighted a delicate balance between model complexity and effectiveness. Some configurations risk overfitting, which is a pertinent issue for future research endeavors. Improving generalization capabilities, without compromising detection accuracy, remains a key area for further investigation.

Looking ahead, several promising avenues for enhancement and innovation present themselves. Enhanced real-time data processing capabilities, particularly for applications in environmental monitoring and industrial control systems, represent a significant area for advancement. Techniques from data assimilation and online learning could be effectively integrated into TransNAS-TSAD to address these challenges. Additionally, the development and implementation of dynamic thresholding strategies, such as the moving average thresholding (MAT) and the incorporation of rolling statistics, offer exciting prospects.

Furthermore, the exploration of hybrid systems that synergize machine learning with simulation approaches, along with advancements in neuro-symbolic systems, could substantially enhance the model's adaptability and effectiveness across various scenarios. Finally, a human-centric approach to machine learning, integrating human feedback in a more intuitive and formalized manner, remains a significant challenge. TransNAS-TSAD stands to benefit greatly from such integration, enhancing not only the interpretability but also the overall usability of the model in real-world applications.

As the landscape of time series anomaly detection continues to evolve, so too will the strategies and methodologies embodied in TransNAS-TSAD, ensuring its continued relevance and efficacy in this dynamic field.

7 Conclusion

TransNAS-TSAD signifies a notable contribution in the realm of time series anomaly detection by merging transformer architecture with neural architecture search and NSGA-II optimization, achieving superior performance across diverse univariate and multivariate datasets. Its robustness in accurately detecting anomalies highlights the effectiveness of this integration, effectively addressing the dual challenge of maintaining accuracy while ensuring computational efficiency-key for practical applications. This framework is further distinguished by its incorporation of advanced adversarial learning paradigms, enabling the detection of nuanced anomalies and marking a significant step forward from traditional methods. As the landscape of data analysis evolves, TransNAS-TSAD not only sets new performance benchmarks in anomaly detection but also opens up exciting avenues for future research, particularly in enhancing real-time processing and integrating human-centric approaches to machine learning. The principles and approaches embodied in TransNAS-TSAD are paving the way for innovative applications across various sectors, shaping the future of machine learning with its groundbreaking methodologies.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant No. EAR 2012123. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government. The work was also supported by the University of Vermont College of Engineering and Mathematical Sciences through the REU program. We would like to extend our gratitude to the providers of the data sets used in this work, whose contributions were invaluable to our research.

Data and code availability This study utilizes several public datasets, namely NAB, UCR, MBA, SMAP, MSL, SWaT, WADI, and SMD, to validate the research findings. Among these, the SWaT and WADI datasets were obtained with permission from the iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design. These datasets are publicly available for research purposes and have been used in accordance with their respective terms of use. The source code, data, and other artifacts are available at https://github.com/ejokhan/TransNAS_TSAD.git.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. Despite the research being supported by the National Science Foundation under Grant No. EAR 2012123 and the University of Vermont College of Engineering and Mathematical Sciences through the REU program, all authors confirm that there are no Conflict of interest to declare.

References

1. Bakumenko A, Elragal A (2022) Detecting anomalies in financial data using machine learning algorithms. Systems 10(5):130. https://doi.org/10.3390/systems10050130
2. Samariya D, Ma J, Aryal S, Zhao X (2023) Detection and explanation of anomalies in healthcare data. Health Infor Sci Syst 11(1):20. https://doi.org/10.1007/s13755-023-00221-
3. Bhanage DA, Pawar AV, Kotecha K (2021) IT infrastructure anomaly detection and failure handling: a systematic literature review focusing on datasets, log preprocessing, machine & deep learning approaches and automated tool. IEEE Access 9:156392–156421. https://doi.org/10.1109/ACCESS.2021.3128283
4. Kammerer K, Hoppenstedt B, Pryss R, Stökler S, Allgaier J, Reichert M (2019) Anomaly detections for manufacturing systems based on sensor data-insights into two challenging real-world production settings. Sensors 19(24):5370. https://doi.org/10.3390/s19245370
5. Brophy E, Wang Z, She Q, Ward T (2023) Generative adversarial networks in time series: a systematic literature review. ACM Comput Surv 55(10):1–31. https://doi.org/10.1145/3559540
6. Li G, Jung JJ (2023) Deep learning for anomaly detection in multivariate time series: approaches, applications, and challenges. Infor Fusion 91:93–102. https://doi.org/10.1016/j.inffus.2022.10.008
7. Thudumu S, Branch P, Jin J, Singh J (2020) A comprehensive survey of anomaly detection techniques for high dimensional big data. J Big Data 7:1–30. https://doi.org/10.1186/s40537-020-00320-x
8. Wang C, Viswanathan K, Choudur L, Talwar V, Satterfield W and Schwan K (2011) Statistical techniques for online anomaly detection in data centers. In 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops (pp. 385–392). IEEE. https://doi.org/10.1109/INM.2011.5990537
9. Leadbetter MR (1991) On a basis for 'Peaks over Threshold' modeling. Stat Probab Lett 12(4):357–362. https://doi.org/10.1016/0167-7152(91)90107-3
10. Ashabi A, Sahibuddin SB, Haghighi MS (2020) Big data: Current challenges and future scope. In: 2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE) (pp. 131–134). IEEE. https://doi.org/10.1109/ISCAIE47305.2020.9108826
11. Al-Sai ZA, Abdullah R (2019) Big data impacts and challenges: a review. In: 2019 IEEE Jordan international joint conference on electrical engineering and information technology (JEEIT) (pp. 150–155). IEEE. https://doi.org/10.1109/JEEIT.2019.8717484
12. Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller PA (2019) Deep learning for time series classification: a review. Data Min Knowl Disc 33(4):917–963. https://doi.org/10.1007/s10618-019-00619-1
13. Ma X, Wu J, Xue S, Yang J, Zhou C, Sheng QZ, Akoglu L (2021) A comprehensive survey on graph anomaly detection with deep learning. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2021.3118815
14. Haq IU, Lee BS, Rizzo DM, Perdrial JN (2023) An Automated Machine Learning Approach for Detecting Anomalous Peak Patterns in Time Series Data from a Research Watershed in the Northeastern United States Critical Zone. arXiv preprint https://doi.org/10.48550/arXiv.2309.07992
15. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN and Polosukhin I (2017) Attention is all you need. Advances in neural information processing systems, 30. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
16. Tuli S, Casale G, Jennings NR (2022) TranAD: Deep transformer networks for anomaly detection in multivariate time series data. Proceed VLDB Endowment 15(6):1201–1214. https://doi.org/10.14778/3514061.3514067
17. Kim J, Kang H, Kang P (2023) Time-series anomaly detection with stacked Transformer representations and 1D convolutional network. Eng Appl Artif Intell 120:105964. https://doi.org/10.1016/j.engappai.2023.105964
18. Arslan F, Javaid A, Awan MDZ (2023) Anomaly Detection in Time Series: Current Focus and Future Challenges. https://doi.org/10.5772/intechopen.111886
19. Elsken T, Metzen JH, Hutter F (2019) Neural architecture search: a survey. J Machine Learn Res 20(1):1997–2017. https://doi.org/10.5555/3322706.3361996
20. Deb K, Pratap A, Agarwal S, Meyarivan TAMT (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197. https://doi.org/10.1109/4235.996017
21. Lu Z, Cheng R, Jin Y, Tan KC, Deb K (2023) Neural architecture search as multiobjective optimization benchmarks: problem formulation and performance assessment. IEEE Trans Evol Comput. https://doi.org/10.1109/TEVC.2022.3233364
22. Lu Z, Deb K, Goodman E, Banzhaf W, Boddeti VN (2020) Nsganetv2: Evolutionary multi-objective surrogate-assisted neural architecture search. In: Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I 16 (pp. 35–). Springer International Publishing. https://doi.org/10.1007/978-3-030-58452-8_3
23. Wei WW (2018) Multivariate Time Series Analysis and Applications. John Wiley & Sons, New Jersey. https://doi.org/10.1109/INM.2011.5990537
24. Liu CL, Hsaio WH, Tu YC (2018) Time series classification with multivariate convolutional neural network. IEEE Trans Industr Electron 66(6):4788–4797. https://doi.org/10.1109/TIE.2018.2864702
25. Lian D, Zheng Y, Xu Y, Lu Y, Lin L, Zhao P, Gao S (2019) Towards fast adaptation of neural architectures with meta learning. In: International Conference on Learning Representations. https://openreview.net/forum?id=r1eowANFvr
26. Liu Y, Sun Y, Xue B, Zhang M, Yen GG, Tan KC (2021) A survey on evolutionary neural architecture search. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2021.3100554
27. Xue Y, Chen C, Słowik A (2023) Neural architecture search based on a multi-objective evolutionary algorithm with probability stack. IEEE Trans Evol Comput. https://doi.org/10.1109/TEVC.2023.3252612
28. Hodge V, Austin J (2004) A survey of outlier detection methodologies. Artif Intell Rev 22:85–126. https://doi.org/10.1023/B:AIRE.0000045502.10941.a9
29. Patcha A, Park JM (2007) An overview of anomaly detection techniques: existing solutions and latest technological trends. Comput Netw 51(12):3448–3470. https://doi.org/10.1016/j.comnet.2007.02.001
30. Lee BS, Kaufmann JC, Rizzo DM, Haq IU (2022) Peak Anomaly Detection from Environmental Sensor-Generated Watershed Time Series Data. In Annual International Conference on Information Management and Big Data (pp. 142–157). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-35445-8_11
31. Landauer M, Onder S, Skopik F, Wurzenberger M (2023) Deep learning for anomaly detection in log data: a survey. Mach Learn Appl 12:100470. https://doi.org/10.1016/j.mlwa.2023.100470

32. Al-amri R, Murugesan RK, Man M, Abdulateef AF, Al-Sharafi MA, Alkahtani AA (2021) A review of machine learning and deep learning techniques for anomaly detection in IoT data. Appl Sci 11(12):5320. https://doi.org/10.3390/app11125320
33. Li S, Jin X, Xuan Y, Zhou X, Chen W, Wang YX, Yan X (2019) Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. Adv Neural Infor Process Syst. https://doi.org/10.5555/3454287.3454758
34. Tang B, Matteson DS (2021) Probabilistic transformer for time series analysis. Adv Neural Infor Process Syst 34:23592–23608
35. Karita S, Chen N, Hayashi T, Hori T, Inaguma H, Jiang Z, Zhang W (2019) A comparative study on transformer vs rnn in speech applications. In: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) (pp. 449–456). IEEE. https://doi.org/10.1109/ASRU46091.2019.9003750
36. Reza S, Ferreira MC, Machado JJM, Tavares JMR (2022) A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks. Expert Syst Appl 202:117275. https://doi.org/10.1016/j.eswa.2022.117275
37. Katrompas A, Ntakouris T, Metsis V (2022) Recurrence and self-attention vs the transformer for time-series classification: a comparative study. In: International Conference on Artificial Intelligence in Medicine (pp. 99–109). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-09342-5_10
38. Ying W, Zheng K, Wu Y, Li J, and Xu X (2020) Neural architecture search using multi-objective evolutionary algorithm based on decomposition. In Artificial Intelligence Algorithms and Applications: 11th International Symposium, ISICA 2019, Guangzhou, China, November 16–17, 2019, Revised Selected Papers 11 (pp. 143–154). Springer Singapore. https://doi.org/10.1007/978-981-15-5577-0_11
39. Borchert O, Salinas D, Flunkert V, Januschowski T, Günnemann S (2022) Multi-objective model selection for time series forecasting. arXiv preprint arXiv:2202.08485. https://doi.org/10.48550/arXiv.2202.08485
40. Chen Y, Meng G, Zhang Q, Xiang S, Huang C, Mu L, Wang X (2019) Renas: Reinforced evolutionary neural architecture search. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4787–4796). https://doi.org/10.1109/CVPR.2019.00492
41. Chu X, Zhang B, Xu R (2020) Multi-objective reinforced evolution in mobile neural architecture search. In European Conference on Computer Vision (pp. 99–113). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-66823-5_6
42. Lu H, Du M, He X, Qian K, Chen J, Sun Y, Wang K (2021) An adaptive neural architecture search design for collaborative edge-cloud computing. IEEE Netw 35(5):83–89. https://doi.org/10.1109/MNET.201.2100069
43. Wang EK, Xu SP, Chen CM, Kumar N (2020) Neural-architecture-search-based multiobjective cognitive automation system. IEEE Syst J 15(2):2918–2925. https://doi.org/10.1109/JSYST.2020.3002428
44. Lu Z, Whalen I, Boddeti V, Dhebar Y, Deb K, Goodman E, Banzhaf W (2019) Nsga-net: neural architecture search using multi-objective genetic algorithm. In: Proceedings of the genetic and evolutionary computation conference (pp. 419–7). https://doi.org/10.1145/3321707.3321729
45. Chitty-Venkata KT, Emani M, Vishwanath V, Somani AK (2022) Neural architecture search for transformers: a survey. IEEE Access 10:108374–108412. https://doi.org/10.1109/ACCESS.2022.3212767
46. Wen Q, Zhou T, Zhang C, Chen W, Ma Z, Yan J, and Sun L (2022) Transformers in time series: A survey. arXiv preprint https://doi.org/10.48550/arXiv.2202.07125
47. Kim YS, Kim MK, Fu N, Liu J, Wang J, Srebric J (2024) Investigating the impact of data normalization methods on predicting electricity consumption in a building using different artificial neural network models. Sustain Cities Soc. https://doi.org/10.1016/j.scs.2024.105570
48. Asesh A (2022) Normalization and bias in time series data. In C. Biele, J. Kacprzyk, W. Kopeć, J. W. Owsiński, A. Romanowski, & M. Sikorski (Eds.), Digital interaction and machine intelligence. MIDI 2021. Lecture notes in networks and systems. (Vol. 440). Springer, Cham. https://doi.org/10.1007/978-3-031-11432-8_8
49. Lima FT, Souza VMA (2023) A large comparison of normalization methods on time series. Big Data Res 34:100407. https://doi.org/10.1016/j.bdr.2023.100407
50. Hundman K, Constantinou V, Laporte C, Colwell I, Soderstrom T (2018) Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 387–395). https://doi.org/10.1145/3219819.3219845
51. Zong B, Song Q, Min MR, Cheng W, Lumezanu C, Cho D, and Chen H (2018) Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In International Conference on Learning Representations. https://openreview.net/forum?id=BJJLHbb0-
52. Su Y, Zhao Y, Niu C, Liu R, Sun W, and Pei D (2019) Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 2828–2837). https://doi.org/10.1145/3292500.3330672
53. Zhang C, Song D, Chen Y, Feng X, Lumezanu C, Cheng W, and Chawla NV (2019) A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 1409–1416). https://doi.org/10.1609/aaai.v33i01.33011409
54. Li D, Chen D, Jin B, Shi L, Goh J, Ng SK (2019) MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In: International conference on artificial neural networks (pp. 703–6). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-30490-4_56
55. Audibert J, Michiardi P, Guyard F, Marti S, Zuluaga MA (2020) Usad: Unsupervised anomaly detection on multivariate time series. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 3395–3404). https://doi.org/10.1145/3394486.3403392
56. Zhang Y, Chen Y, Wang J, Pan Z (2021) Unsupervised deep anomaly detection for multi-sensor time-series signals. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2021.3102110
57. Deng A, Hooi B (2021, May) Graph neural network-based anomaly detection in multivariate time series. In: Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 5, pp. 4027–4035). https://doi.org/10.1609/aaai.v35i5.16523
58. Ahmad S, Lavin A, Purdy S, Agha Z (2017) Unsupervised real-time anomaly detection for streaming data. Neurocomputing 262:134–147. https://doi.org/10.1016/j.neucom.2017.04.070
59. Nakamura T, Imamura M, Mercer R, Keogh E (2020) Merlin: Parameter-free discovery of arbitrary length anomalies in massive time series archives. In: 2020 IEEE international conference on data mining (ICDM) (pp. 1190–95). IEEE. https://doi.org/10.1109/ICDM50108.2020.00147
60. Dau HA, Bagnall A, Kamgar K, Yeh CCM, Zhu Y, Gharghabi S, Keogh E (2019) The UCR time series archive. IEEE/CAA J

Automatica Sinica 6(6):1293–1305. https://doi.org/10.1109/JAS.2019.1911747
61. Moody GB, Mark RG (2001) The impact of the MIT-BIH arrhythmia database. IEEE Eng Med Biol Mag 20(3):45–50. https://doi.org/10.1109/51.932724
62. Boniol P, Linardi M, Roncallo F, Palpanas T (2020) Automated anomaly detection in large sequences. In 2020 IEEE 36th international conference on data engineering (ICDE) (pp. 1834–1837). IEEE. https://doi.org/10.1109/ICDE48307.2020.00182
63. Mathur AP, Tippenhauer NO (2016) SWaT: A water treatment testbed for research and training on ICS security. In: 2016 international workshop on cyber-physical systems for smart water networks (CySWater) (pp. 31–). IEEE. https://doi.org/10.1109/CySWater.2016.7469060
64. Ahmed CM, Palleti VR, Mathur AP (2017) WADI: a water distribution testbed for research in the design of secure cyber physical systems. In Proceedings of the 3rd international workshop on cyber-physical systems for smart water networks (pp. 25–28). https://doi.org/10.1145/3055366.3055375
65. Johnson TB, Guestrin C (2018) Training deep models faster with robust, approximate importance sampling. Advances in Neural Information Processing Systems 31. https://doi.org/10.5555/3327757.3327829
66. Nokhwal S, Chilakalapudi P, Donekal P, Chandrasekharan M, Nokhwal S, Swaroop R, Chaudhary A (2023) Accelerating neural network training: A brief review. arXiv preprint https://doi.org/10.48550/arXiv.2312.10024
67. Coquelin D, Debus C, Götz M, von der Lehr F, Kahn J, Siggel M, Streit A (2022) Accelerating neural network training with distributed asynchronous and selective optimization (DASO). J Big Data 9(1):14. https://doi.org/10.1186/s40537-021-00556-1