# Position: Quo Vadis, Unsupervised Time Series Anomaly Detection?

**M. Saquib Sarfraz** [1 2] **Mei-Yen Chen** [1] **Lukas Layer** [1] **Kunyu Peng** [2] **Marios Koulakis** [2]

## Abstract

The current state of machine learning scholarship in Timeseries Anomaly Detection (TAD) is plagued by the persistent use of flawed evaluation metrics, inconsistent benchmarking practices, and a lack of proper justification for the choices made in novel deep learning-based model designs. Our paper presents a critical analysis of the status quo in TAD, revealing the misleading track of current research and highlighting problematic methods, and evaluation practices. **Our position advocates for a shift in focus from solely pursuing novel model designs to improving benchmarking practices, creating non-trivial datasets, and critically evaluating the utility of complex methods against simpler baselines**. Our findings demonstrate the need for rigorous evaluation protocols, the creation of simple baselines, and the revelation that state-of-the-art deep anomaly detection models effectively learn linear mappings. These findings suggest the need for more exploration and development of simple and interpretable TAD methods. The increment of model complexity in the state-of-the-art deep-learning based models unfortunately offers very little improvement. We offer insights and suggestions for the field to move forward.

## 1. Introduction

Time series anomaly detection (TAD) is an active field of machine learning with applications across multiple industries. For instance, many real-world systems such as vehicles, manufacturing plants, robots, and patient monitoring systems, involve a large number of interconnected sensors producing a great amount of data over time that can be used to detect anomalous behaviour. The anomalies can manifest as single irregular points or groups of such points whose interpretation as anomalous might depend on the system's operational history or on the inter-connectivity among sub-modules.

Given the complexity of the problem and inspired from the successes in other areas, such as natural language or audio processing, many state-of-the-art deep-learning architectures have been adjusted and applied to it. Such approaches aim to learn a latent representation of the normal time-series data, e.g. LSTM (Park et al., 2017), Transformer (Tuli et al., 2022; Xu et al., 2022), and sometimes explicitly model the inter-dependency among the sub-components in the system, e.g. graph neural networks (Deng & Hooi, 2021; Chen et al., 2021). Based on the assumption that the anomalies constitute unseen patterns which will not be modelled during reconstruction of the series from the model, the difference between the original and reconstructed series is used to detect them.

Although it is well intended, this line of research has never provided evidence of the necessity of deep-learning, which has been challenged namely in Audibert et al. (2022). The state-of-the-art (SOTA) deep-learning approaches proceeded to introduce models of increased complexity using questionable validation processes. Those processes involve unsuitable benchmark datasets (Wu & Keogh, 2022) and, most harmful to this field, the use of flawed evaluation protocols (Kim et al., 2022). The protocol which introduced the most pitfalls is the point adjustment (PA) applied on the point-wise F1 score which practically favors noisy predictions. It was gradually introduced in a series of papers (Xu et al., 2018; Audibert et al., 2020; Shen et al., 2020; Su et al., 2019c) with the original intention of calibrating the anomaly detection threshold on a hold-out dataset, but it was subsequently demonstrated in Kim et al. (2022) that uniformly random predictions outperform SOTA methods and their performance tends to one as the average length of the anomalies increases. Although using the standard F1 score without point-adjust avoids those pitfalls, it still leaves a gap by only focusing on point-wise time-stamp level detection versus anomaly instance level detection, which led to the introduction of new complementary range-based metrics such as the ones in Tatbul et al. (2018), Wagner et al. (2023).

The goal of this paper is to guide the TAD community to-

[1]Mercedes-Benz Tech Innovation, Ulm, Germany [2]Karlsruhe Institute of Technology, Karlsruhe, Germany. Correspondence to: M. Saquib Sarfraz <saquibsarfraz@gmail.com>.

*Proceedings of the $41^{st}$ International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. 

wards more meaningful progress through rigorous benchmarking practices and a focus on studying the utility of their models by drawing useful but simple baselines. We achieve this with the following contributions: 1.) We introduce simple and effective baselines and demonstrate that they perform on par or better than the SOTA methods, thus challenging the efficiency and effectiveness of increasing model complexity to solve TAD problems. 2.) We reinforce this position by reducing trained SOTA models to linear models which are distillations of them but still perform on par. Thus from the point of view of the TAD task on the current datasets, those models perform roughly a linear separation of the anomalies from the nominal data.

Our code[1] is available on GitHub to easily run the baselines and benchmarks.

## 2. Related Work

Anomaly detection in time series data has been extensively studied, with methods ranging from univariate to multivariate and including complex deep-learning models (Li et al., 2019; Zhang et al., 2019; Zhao et al., 2020; Su et al., 2019a; Zong et al., 2018; Hundman et al., 2018a; Deng & Hooi, 2021; Chen et al., 2021). These models are trained to forecast or reconstruct presumed normal system states and then deployed to detect anomalies in unseen test datasets. The anomaly score defined as the magnitude of prediction or reconstruction errors serves as an indicator of abnormality at each time stamp. Model performance is often evaluated as a binary classification problem, with the anomaly scores thresholded into binary labels. A comprehensive review of anomaly detection methods can be found in (Schmidl et al., 2022; Blázquez-García et al., 2021).

**Classical machine learning methods:** A basic approach to anomaly detection in time-series data involves treating sample points of each sensor as independent and using classical statistical methods on the individual univariate series. For instance, regression models are used for the prediction from other sensor measurements (Salem et al., 2014). Principal Component Analysis (PCA) is utilized for dimensionality reduction and reconstruction (Shyu et al., 2006). Other methods for anomaly detection on time series data take temporal dependency or correlation among sensors into account. These include modeling families of hidden Markov chains (Patcha & Park, 2007) or graph theory (Boniol et al., 2020). Signal transformation (Kanarachos et al., 2017), isolation forest (Bandaragoda et al., 2018; Liu et al., 2008), Auto-Regressive Integrated Moving Average (ARIMA) (Yaacob et al., 2010) and clustering (Angiulli & Pizzuti, 2002; Boniol et al., 2021; Tran et al., 2020). Time-series discord discovery has recently emerged as a favored choice for univariate data analysis. A recent method MERLIN (Nakamura et al., 2020) is considered to be the state-of-the-art for univariate anomaly detection, as it iteratively varies the length of a subsequence and searches for those that are greatly different from their nearest neighbors as candidates of abnormality. Also see (Paparrizos et al., 2022) for a comprehensive performance comparison of different classical TAD methods on univariate data.

**Deep learning methods:** Anomaly in time series might be hidden in peculiar dependencies among sub-modules in a system or over its operation history that are hard to detect with manual feature engineering. Modern deep-learning models that can learn temporal dependency via recursive networks (e.g. LSTM) or attention mechanisms (e.g. Transformer) or by explicitly representing the correlation among sensors (e.g. Graph Neural Networks) have been proposed as the cutting-edge methods for TAD. For instance, LSTM-VAE (Park et al., 2017) used a variational autoencoder that is based on LSTM and reconstructs the test data with variational inferences. DAGMM (Zong et al., 2018) utilized deep autoencoders and Gaussian mixture model to jointly model a low-dimensional representation which is then used to reconstruct each time stamp. It computes the reconstruction error for anomaly detection. OmniAnomaly(Su et al., 2019a) modeled the time series data as stochastic random process with variational autoencoders (VAE) and established reconstruction likelihood as an anomaly score. Another approach, USAD (Audibert et al., 2020), introduced a two-phase training paradigm in which two autoencoders and two decoders are trained under the adversarial game-style. Among the more recent methods that currently represent the state-of-the-art deep models on anomaly detection are GDN (Deng & Hooi, 2021) and TranAD (Tuli et al., 2022). GDN (Deng & Hooi, 2021) models the inter-connectivity among sensors as a graph and used graph attention network to forecast the sensor measurement. The deviation between true observation and model predictions is then used to quantify anomalies. TranAD (Tuli et al., 2022) is a transformer based approach that proposed a new transformer architecture for anomaly detection. It introduced several components with a two transformer-based encoder and decoders using multi-head attention blocks. The approach then proposed a two-phase training scheme utilizing adversarial and meta learning procedures. Another recent transformer based approach Anomaly Transformer (Xu et al., 2022) introduced a new attention block and a min-max loss which helps learn two separate series associations, one prior which aims to capture local associations which in cases of anomaly would be caused by the continuity around it and series associations which should encode deeper information about the temporal context. Overall both methods results in complicated schemes. A similar approach in designing a transformer

[1] Code: https://github.com/ssarfraz/QuoVadisTAD

based model along with meta learning objectives and optimal transport has been presented in (Li et al., 2023).

Aside from the anomaly detection approaches, many efforts has been put in creating useful anomaly detection benchmarks. Some recent studies, for instance (Wu & Keogh, 2022) have shown how some of these datasets suffer from potential flaws, such as triviality, unrealistic density of anomaly, or mislabeling.

## 3. Methods

Among the numerous anomaly detection approaches presented in the past, there is often something consistent - they tend to overlook simpler baselines in pursuit of novelty. This leads to overly complex engineered solutions without much utility and a good rationale. Towards this end, we propose simple methods that exceed the performance of current best-published anomaly detection approaches. As a result, these baselines help us to understand the complexity of the underlying problem and provide a solid foundation for further investigation. Of note, our contribution is properly setting up these known methods and creating a set of strong baselines.

### 3.1. Preliminaries

We introduce some notations which are used to formally define the task of unsupervised TAD and describe the methods used. The training data consist of a time series $\mathbf{X} = [\mathbf{x}_1, \dots \mathbf{x}_T] \in \mathbb{R}^{T \cdot F}$ which only contains non-anomalous timestamps. Here $T$ is the number of timestamps and $F$ the number of features. The test set, $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1, \dots \hat{\mathbf{x}}_{\hat{T}}] \in \mathbb{R}^{\hat{T} \cdot F}$ contains both normal and anomalous timestamps and $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_{\hat{T}}] \in \{0, 1\}^{\hat{T}}$ represents their labels, where $\hat{y}_t = 0$ denotes a normal and $\hat{y}_t = 1$ an anomalous timestamp $t$. Then the task of anomaly detection is to select a function $f_{\boldsymbol{\theta}} : \mathbf{X} \to \mathbb{R}$ such that $f_{\boldsymbol{\theta}}(\mathbf{x}_t) = \tilde{y}_t$ estimates the anomaly value $\hat{y}_t$[2]. The (potentially empty) set of parameters $\boldsymbol{\theta}$ is estimated using the training data $\mathbf{X}$. In most methods, usually an intermediate error vector function $err_{\boldsymbol{\theta}} : \mathbf{X} \to \mathbb{R}^F$ is estimated which computes vectors representing an error along all sensors, we also denote by $\mathbf{E} = err_{\boldsymbol{\theta}}(\hat{\mathbf{X}})$ the predicted test error vectors.

The error vectors $\mathbf{E}$ estimated from any of the methods provide a measure of the deviation of the test features from normality. Normalization of error vectors sometimes is necessary before detecting anomalies due to variations in error behavior across sensors. Two normalization methods are often used: scaling using robust statistics such as median and inter-quartile range (Deng & Hooi, 2021) and scaling using mean and standard deviation. The choice of normalization approach can impact anomaly detection accuracy, and careful consideration should be given to the selected method. The impact of error vector normalization on datasets is demonstrated through an ablation study in section 4.4. Once the error vectors are normalized, the final output is a measure of the vector sizes. Given that we are working on the anomaly detection scenario, the most fitting metric is $L^\infty$ which computes the largest absolute error between the different sensors, $\|\mathbf{e}_t\|_\infty = \max_{i \leq F}\{|e_t^i|\}$.

[2]The range of $\tilde{y}_t$ values may differ from $\hat{y}_t \in 0, 1$, necessitating thresholding before obtaining actual predictions. Typically, the threshold which yields the best score on the training or validation data is selected.

### 3.2. Proposed simple and effective baselines

**Sensor range deviation:** The range of sensor values observed during normal operation can be useful in identifying out-of-distribution (OOD) samples. Anomalies in time series data can occur when the sensor values deviate from their usual range. Therefore, if the sensor values in a test data point fall outside the observed range, it may indicate the presence of an anomaly. Formally this is defined as:

$$f(\hat{\mathbf{x}}_t) = \left\{ \begin{array}{ll} 0 & if\ \hat{\mathbf{x}}_t \in [\min(\mathbf{X}), \max(\mathbf{X})] \\ 1 & otherwise \end{array} \right\}$$

This represents a minimum level of detection performance that any advanced method should be able to surpass.

**L2-norm: Magnitude of the observed time stamp:** In the case of multivariate time series data, the magnitude of the vector at a particular timestamp may serve as a relevant statistic for detecting OOD samples. This can be easily computed by taking the L2-norm of the vector, thus $f(\hat{\mathbf{x}}_t) = \|\hat{\mathbf{x}}_t\|_2$. By using the magnitude as an anomaly score, we have discovered that it can be an effective and robust baseline for identifying anomalies in multivariate datasets.

**NN-distance: Nearest neighbor distance to the normal training data:** A sample that deviates from normal data should have a greater distance from it. Therefore, using the nearest-neighbor distance between each test time-stamp and the train data as an anomaly score can serve as a reliable baseline. In fact, in many cases, this method outperforms several state-of-the-art techniques.

**PCA reconstruction error:** Our simplest reconstruction method can be seen as an outlier detection on a lower dimensional linear approximation of the train dataset single timestamp features.

After centering the training set $\mathbf{X}$ on its mean, using PCA, we compute the principal components of its features. This defines an affine approximation of $\mathbf{X}$ centered on the origin which can be expressed by the eigenvector matrix $\mathbf{U} \in \mathbb{R}^{F \cdot F'}$, where $F' < F$ is a fixed number of the first principle components. Then the test set $\hat{\mathbf{X}}$ is transformed to

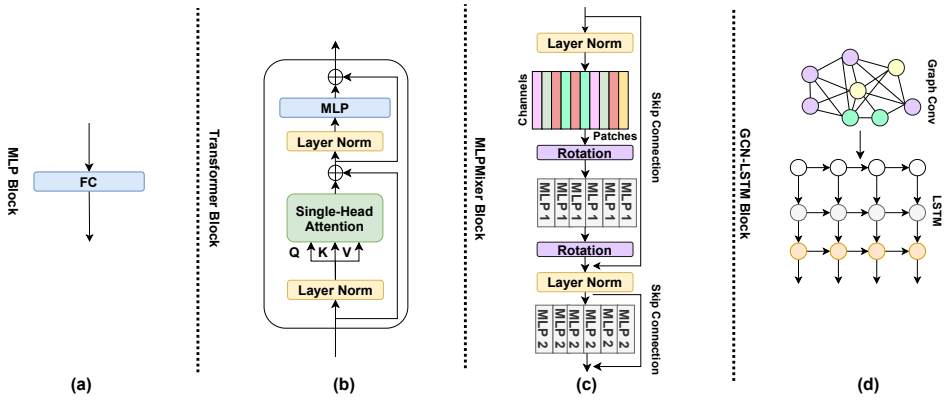



*Figure 1.* Proposed simple neural-network baselines

$\tilde{\mathbf{X}} = \hat{\mathbf{X}}\mathbf{U}^T\mathbf{U} \in \mathbb{R}^{\hat{T}\cdot F}$ and we consider the reconstruction error vectors $\mathbf{E} = err_{\mathbf{U}}(\hat{\mathbf{X}}) = \hat{\mathbf{X}} - \tilde{\mathbf{X}}$.

There are two ways to interpret this transform. The first one is as a linear reconstruction of the test data, which is equivalent to using a linear autoencoder trained with the mean squared error loss on the training set, see (Bourlard & Kamp, 1988) and (Baldi & Hornik, 1989). The second way is to interpret it as the projection of each vector of $\hat{\mathbf{X}}$ to the linear subspace $\mathcal{S} = \text{span}(\text{cols}(\mathbf{U})) \subset \mathbb{R}^F$ formed by the principal components in $\mathbf{U}$. This interpretation highlights the linearity and simplicity of the method as each error vector $\mathbf{e}_t$ connects $\mathbf{x}_t$ with $\mathcal{S}$ and is perpendicular to $\mathcal{S}$, thus expresses the distance between $\mathbf{x}_t$ and $\mathcal{S}$.

### 3.3. Proposed neural network blocks baselines

Contemporary anomaly detection techniques based on deep learning utilize modern neural networks to create solutions with varying levels of sophistication. Among the commonly employed architectures are auto-encoders (AE), long short-term memory (LSTM) networks, multi-layer perceptrons (MLPs), graph convolution networks (GCN), and Transformers. These neural network structures serve as the foundational components for designing intricate models intended for anomaly detection. In order to provide context for the usefulness of the more elaborate solutions, we utilize these architectures in their most basic form as a set of baselines. It is reasonable to expect that any solution which employs these as foundational components should perform better, provided they are trained on rich enough datasets of normal examples. Our experiments demonstrate that, in most cases, these basic baselines perform better than models that incorporate a combination of these structures for the purpose of anomaly detection. Therefore, establishing such baselines may help understand the rationale behind the development of more complex models.

**1-layer linear MLP as auto-encoder:** As the first simplest neural baseline we use a single hidden-layer MLP without any activation as an auto-encoder.

**Single block MLP-Mixer:** Among the more modern variants of MLPs, the MLPMixer (Tolstikhin et al., 2021) has been shown to perform quite well on many vision problems. The architecture includes several MLP layers, called MLP-Mixer blocks. Each MLP-Mixer block consists of two sub-layers: a token-mixing sub-layer and a channel-mixing sub-layer. These operate on the spatial dimension and the channel dimension of the input feature maps. The entire architecture consists of stacking several MLP-Mixer blocks, allowing the network to capture increasingly complex spatial and cross-channel dependencies in the input. We include a single standard block of MLPMixer as our baseline.

**Single Transformer block:** Since transformers are increasingly used in several recent anomaly detection methods, we use a basic transformer block with one single-head attention and one fully connected layer as a feed-forward output. This serves as the simplest and basic single transformer block baseline.

**1-layer GCN-LSTM block:** Using a single GCN layer feeding into a LSTM layer is a simple yet effective baseline for learning graph structure on multivariate time series data. The GCN layer is used to model the relationships between different time series variables, while the LSTM layer is used to capture temporal dependencies within each time series variable. The output of the LSTM layer is then forwarded to the output regression layer directly. Overall, this baseline provides a basic framework for jointly modeling the graph structure and temporal dependencies in multivariate time series data. Many recently published methods extend and improve upon this by incorporating additional GCN or LSTM layers, using attention mechanisms, or incorporating other types of graph neural networks.

Figure 1 illustrates the proposed baseline neural network

blocks. These baseline models are trained and compared in both reconstruction and forecasting modes.

### 3.4. Univariate time series representation

Univariate time series data consist of a single observation at each timestamp, and most deep-learning methods designed for multivariate data are not directly applicable. Consequently, the most effective approaches for analyzing univariate data are typically focused on identifying unusual subsequences, or discords, within the time series. State-of-the-art discord discovery methods, for instance (Nakamura et al., 2020), focus on optimizing the complexity and parameters of such methods that typically involve comparing windowed distances between timestamps. In this work, we use a similar yet effective representation for univariate time series data that allows the discovery of anomalies. Specifically, we represent each timestamp as a vector in $\mathbb{R}^{w+1}$, where $w$ denotes the number of preceding time stamps. This representation can be efficiently computed in a sliding window fashion and has linear time complexity, making it efficient for practical use. In section A.2.1, we demonstrate that the impact of the window size on performance is relatively low and a small fixed window of $w = 4$ suffice for the considered univariate datasets.

### 3.5. Evaluation metrics

A lot of papers introduced and criticised different metrics. In our view, anomaly detection shares a lot with object detection and semantic segmentation in computer vision, therefore it would need two metrics to fully capture model performance. The point-wise which captures the quality of the detection of individual anomalies and range-wise which expresses the quality of the anomaly segmentation. For the point-wise anomaly detection, we use the standard **F1 score**, which actually equals to the 1-dimensional Dice coefficient. For completeness, we also include the flawed and commonly used **F1 score with point adjustment** denoted as $\mathbf{F1_{PA}}$. For the range-wise metrics, we followed the work in this direction starting with the **Time-series precision and recall metrics** defined in (Tatbul et al., 2018) and then corrected for bias in (Wagner et al., 2023) and we use the latter to compute an F1 score denoted as $\mathbf{F1_T}$.

Below are the definitions of the three scores we use together with the corresponding testing protocols:

**F1**: Let $[\hat{y}_1, \dots, \hat{y}_{\hat{T}}]$ be the ground truth per time-stamp on the test set and $[\tilde{y}_1^{thr}, \dots, \tilde{y}_{\hat{T}}^{thr}]$ the corresponding predictions set to 1 when $\tilde{y}_i > thr$ else to 0. The hits are defined as $TP^{thr} = |\{i \leq \hat{T} \mid \tilde{y}_i^{thr} = \hat{y}_i\}|$, $FP^{thr} = |\{i \leq \hat{T} \mid \tilde{y}_i^{thr} = 1 \; and \; \hat{y}_i = 0\}|$ and $FN^{thr} = |\{i \leq \hat{T} \mid \tilde{y}_i^{thr} = 0 \; and \; \hat{y}_i = 1\}|$. Then the precision $Prec^{thr}$, recall $Rec^{thr}$ and F1-score $F1^{thr}$ are defined as usual based on those values. The final score is then $F1 = \max_{thr \in \mathbb{R}} F1^{thr}$.

$\mathbf{F1_{PA}}$: The final F1 score is computed exactly as before. This metric is different in its evaluation protocol which adjusts the predictions using the ground truth. Namely, for every contiguous anomaly interval $A = [t_1, \dots, t_2]$ in the ground truth, if there is at least one $i \in A$ such that $\tilde{y}_i = 1$, then for every $j \in A$, $\tilde{y}_j$ is set to 1. In other words, if an anomaly interval is hit once by the predictions, then all predictions in the interval are corrected to match the ground truth.

$\mathbf{F1_T}$: Let $\mathcal{A}, \mathcal{P}$ be respectively the set of all ground truth and prediction anomaly intervals. Also let $\mathcal{P}_A = \{P \in \mathcal{P} \mid |A \cap P| > 0\}$ be the prediction intervals intersected by $A$. Then precision and recall are defined as follows:

$$Prec_T(\mathcal{A}, \mathcal{P}) = \frac{1}{|\mathcal{P}|} \sum_{P \in \mathcal{P}} \gamma(|\mathcal{A}_P|, P) \frac{|\bigcup \mathcal{A} \cap P|}{|P|}$$

$$Rec_T(\mathcal{A}, \mathcal{P}) = \frac{1}{|\mathcal{A}|} \sum_{A \in \mathcal{A}} \gamma(|\mathcal{P}_A|, A) \frac{|\bigcup \mathcal{P} \cap A|}{|A|}$$

The above definition is consistent with both (Tatbul et al., 2018) and (Wagner et al., 2023). The full formula in the latter paper for recall is

$$Rec_T(\mathcal{A}, \mathcal{P}) = \frac{1}{|\mathcal{A}|} \sum_{A \in \mathcal{A}} [\alpha \mathbb{1}(|\mathcal{P}_A| > 0)$$

$$+(1-\alpha)\gamma(|\mathcal{P}_A|, A) \sum_{P \in \mathcal{P}} \frac{\sum_{t \in P \cap A} \delta(t - \min A, |A|)}{\sum_{t \in A} \delta(t - \min A, |A|)}],$$

where $0 \leq \alpha \leq 1$, $\delta \geq 1$ and $Prec_T(\mathcal{A}, \mathcal{P}) = Rec_T(\mathcal{P}, \mathcal{A})$. (Wagner et al., 2023) proposed to fix the parameters $\alpha, \delta$ to 0 and a constant function, in order to derive their formula for $\gamma$.

Under this assumption, we simplified those formulas to make them more comprehensible. Here we use the corrected $\gamma(n, A) = (\frac{|A|-1}{|A|})^{n-1}$ which guarantees that recall is increasing relative to the threshold of the anomaly detector. To provide some intuition, e.g. the recall computes an average of the fraction of ground truth intervals overlapped by the prediction which expresses the amount of discovery success. Every term is weighted though by $\gamma$ which decreases in value as multiple predictions hit the same ground truth interval, thus penalizing duplicates. Note that $Prec_T(\mathcal{A}, \mathcal{P}) = Rec_T(\mathcal{P}, \mathcal{A})$, i.e. precision measures the recall of prediction intervals by the ground truth. Finally, the F1-score denoted by $F1_T$ is defined as usual using $Prec_T$ and $Rec_T$.

| Dataset | Sensors (traces) | Train | Test | #Anomalies (%) |
|---|---|---|---|---|
| UCR/IB-16 | 1 | 1200 | 6301 | 12 (0.19%) |
| UCR/IB-17 | 1 | 1600 | 5900 | 111 (1.88%) |
| UCR/IB-18 | 1 | 2300 | 5200 | 102 (1.96%) |
| UCR/IB-19 | 1 | 3000 | 4500 | 10 (0 .22%) |
| SWaT | 51 | 47520 | 44991 | 4589 (12.20%) |
| WADI-127 | 127 | 118750 | 17280 | 1633 (9.45%) |
| WADI-112 | 112 | 118750 | 17280 | 918 (5.31%) |
| SMD | 38 (28) | 25300 | 25300 | 1050 (4.21%) |

*Table 1.* The statistical profile of the datasets in the experiment.

The F1 scores are calculated using the best threshold computed on the test dataset and this threshold is also used to compute the corresponding precision and recall. Though we are not content with the threshold tuning, we choose this in order to follow the same protocol used in the published methods we have included for comparison. Here, it is important to also include the Area Under the Precision Recall Curve (AUPRC) metric instead of only the F1 score obtained with an optimal threshold. AUPRC provides a more realistic estimation of how well a method would perform in practical settings, where an estimated threshold based on a hold-out set would be used. In our appendix, we include tables ( 9, 10, 11, 12) with the separate precision, recall, and AUPRC values.

## 4. Analysis

**Time series datasets:** Overall, we used six commonly used benchmark datasets in our study. Here, we report the details (Table 1) and results from three multivariate datasets (SWaT, WADI, and SMD) and four univariate datasets (UCR/Internal Bleeding). The other two commonly used multivariate datasets (SMAP and MSL) have been identified in (Wu & Keogh, 2022) as potentially flawed containing trivial and unrealistic density of anomalies. For completeness, the descriptions and results of these two datasets are included in the appendix section A.3.

Univariate HexagonML (UCR) datasets - InternalBleeding (IB) (Guillame-Bert & Dubrawski, 2017): contains four univariate traces as the vital signs (arterial blood pressure). The anomalies are synthetic by adding a series of sine waves to one cycle or by injecting random numbers to a certain segment (Figure 2). The unique and well-controlled anomalies in each trace allow a clean and sound evaluation among different approaches (Wu & Keogh, 2022).

Secure Water Treatment (SWaT) (Mathur & Tippenhauer, 2016) and Water Distribution (WADI) (Ahmed et al., 2017) datasets: contain sensor measurements of a water treatment test-bed. Although SWaT is commonly used as a benchmark in recent publications, it should be noted that its use as a benchmark should be discontinued as it is flawed and unreliable Eamonn Keogh (personal communication, 7 May, 2024), see also (Wagner et al., 2023). The WADI dataset demonstrates the inconsistency in reporting performance comparisons in the TAD literature. The complete set of WADI contains 127 sensors (denoted as WADI-127 in our study). However, some recent methods (Tuli et al., 2022; Deng & Hooi, 2021; Kim et al., 2022; Chen et al., 2021; Feng & Tian, 2021) use a specific subset of sensors when making comparisons without specifying the exact used sensors nor the reasons for such selection. Furthermore, in many cases, the selected subsets are inconsistent among competing methods. In order to provide a fair overview of this impact on performance, we conducted our experiments on all the 127 WADI sensors (denoted as WADI-127) and on the subset of 112 sensors used in some recent studies (Deng & Hooi, 2021) (denoted as WADI-112), separately.

Server Machine Dataset (SMD) (Su et al., 2019c): contains 38 sensors from 28 machines for 10 days. Table 1 reports the average length of each trace. Following the protocol, all models are trained on each machine separately and the results are averaged from 28 different models.

**Evaluation:** We evaluate several state of the art representative deep learning based methods on commonly used time-series benchmarks. To clearly show their utility, we evaluate these 1). under point-adjust $\mathbf{F1_{PA}}$ which is the common metric increasingly used in recent proposals. 2.) standard point-wise $\mathbf{F1}$ and 3.) Time-series range-wise metric $\mathbf{F1_T}$. See section 3.5 for the definitions. To highlight the prevalent use of flawed point-adjust $\mathbf{F1_{PA}}$, similar to (Kim et al., 2022), we also evaluate a random prediction:

**Random:** The $\mathbf{F1_{PA}}$ protocol considers the whole interval of an anomaly as correctly predicted, as soon as the prediction considers a single point of the interval as anomalous. The random prediction directly shows that, under the point-adjust evaluation, methods might achieve high scores just because they have very noisy outputs. In the random baseline setting, each timestamp is predicted anomalous with probability 0.5 and we report the score achieved over five independent runs.

### 4.1. Model setup

In this section we summarize our data preprocessing steps and the hyperparameters used to train the models. The features were scaled to the interval $[0, 1]$ in the training dataset and the learned scaling parameters were used to scale the testing dataset. For all of our NN baselines, when trained in forecasting mode, we used a time window of size 5. We used a 90/10 split to make the train and the validation set. The validation set is only used for early stopping to avoid over-fitting and the Adam optimizer with learning rate 0.001 and a batch size of 512 were used.

**PCA reconstruction error**: For multivariate data, this method uses the first 30 principal components when data

| | | SWaT | | | WADI_127 | | | WADI_112 | | | SMD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $F1_{PA}$ | $F1$ | $F1_T$ | $F1_{PA}$ | $F1$ | $F1_T$ | $F1_{PA}$ | $F1$ | $F1_T$ | $F1_{PA}$ | $F1$ | $F1_T$ |
| **SOTA methods** | MERLIN (Nakamura et al., 2020) | 0.934 | 0.217 | 0.286 | 0.560 | 0.335 | 0.354 | 0.699 | 0.473 | 0.503 | 0.886 | 0.384 | 0.473 |
| | DAGMM (Zong et al., 2018) | 0.830 | 0.770 | 0.402 | 0.363 | 0.279 | 0.406 | 0.829 | 0.520 | 0.609 | 0.840 | 0.435 | 0.379 |
| | OmniAnomaly (Su et al., 2019b) | 0.831 | 0.773 | 0.367 | 0.387 | 0.281 | 0.410 | 0.742 | 0.441 | 0.496 | 0.804 | 0.415 | 0.353 |
| | USAD (Audibert et al., 2020) | 0.827 | 0.772 | 0.413 | 0.375 | 0.279 | 0.406 | 0.778 | 0.535 | 0.573 | 0.841 | 0.426 | 0.364 |
| | GDN (Deng & Hooi, 2021) | 0.866 | 0.810 | 0.385 | 0.767 | 0.347 | 0.434 | 0.833 | 0.571 | 0.588 | **0.929** | 0.526 | 0.570 |
| | TranAD (Tuli et al., 2022) | 0.865 | 0.799 | 0.425 | 0.671 | 0.340 | 0.353 | 0.680 | 0.511 | 0.589 | 0.827 | 0.457 | 0.390 |
| | AnomalyTransformer (Xu et al., 2022) | 0.941 | 0.765 | 0.331 | 0.560 | 0.209 | 0.219 | 0.817 | 0.503 | 0.555 | 0.923 | 0.426 | 0.351 |
| **Simple baselines** | Random | **0.963** | 0.218 | 0.217 | **0.783** | 0.101 | 0.106 | **0.907** | 0.101 | 0.106 | 0.894 | 0.080 | 0.080 |
| | Sensor range deviation | 0.234 | 0.231 | 0.230 | 0.129 | 0.101 | 0.098 | 0.632 | 0.465 | 0.526 | 0.297 | 0.132 | 0.116 |
| | L2-norm | 0.847 | 0.782 | 0.366 | 0.353 | 0.281 | 0.410 | 0.749 | 0.513 | 0.607 | 0.799 | 0.404 | 0.338 |
| | 1-NN distance | 0.847 | 0.782 | 0.372 | 0.372 | 0.281 | 0.410 | 0.751 | 0.568 | 0.618 | 0.833 | 0.463 | 0.384 |
| | PCA Error | 0.895 | **0.833** | **0.574** | 0.621 | **0.501** | **0.557** | 0.783 | **0.655** | **0.699** | 0.921 | **0.572** | **0.580** |
| **NN base-lines** | 1-Layer MLP | 0.856 | 0.771 | 0.519 | 0.295 | 0.267 | 0.384 | 0.601 | 0.502 | 0.558 | 0.829 | 0.514 | 0.487 |
| | Single block MLPMixer | 0.865 | 0.780 | 0.549 | 0.335 | 0.275 | 0.396 | 0.597 | 0.497 | 0.552 | 0.819 | 0.512 | 0.472 |
| | Single Transformer block | 0.854 | 0.787 | 0.526 | 0.471 | 0.289 | 0.416 | 0.646 | 0.534 | 0.575 | 0.781 | 0.489 | 0.420 |
| | 1-Layer GCN-LSTM | 0.905 | 0.829 | 0.532 | 0.593 | 0.439 | 0.540 | 0.748 | 0.596 | 0.645 | 0.847 | 0.550 | 0.535 |

*Table 2.* Experimental results for SWaT, WADI, and SMD datasets. The bold and underline marks the best and second-best value. $F1_{PA}$: F1 score with point-adjust; $F1$: the standard point-wise F1 score; $F1_T$: time-series range-wise F1 score

has more than 50 sensors and 10 otherwise. On univariate datasets, the first 2 principal components with a window size of 5 are used.

**1-layer Linear MLP**: A hidden layer of size 32 is used.

**Single block MLP-Mixer** and **Single Transformer block** both use an embedding of 128 for the hidden layer.

**1-layer GCN-LSTM block**: The dimension for the GCN output nodes is set to 10 and for LSTM layer to 64 units.

Our neural network baselines are trained in the forecasting mode, similar to most other methods we are comparing with. We also provide their performance for the reconstruction mode in the appendix section A.2.2.

**Hyperparameter sensitivity:** Most of the simple baselines don't have tunable hyperparameters. The only exceptions are the projection dimension of the PCA method and the sliding window for univariate series. We have included their ablations in sections 4.5 and A.2.1. We trained our neural network baseline models using the same hyperparameters as stated above on all multivariate datasets. The purpose of this analysis was to demonstrate that even with basic hyperparameters, these simple neural networks can achieve comparable performance to SOTA deep learning models. The fact that the hyperparameters of the SOTA models were optimized for each respective datasets, while the simple NN baseline models used the same set of hyperparameters, highlights less reliance on dataset-specific tuning.

**Published SOTA methods**: All methods were trained with the hyper-parameters recommended in their respective papers, where possible, with their official implementations or the implementations provided in (Tuli et al., 2022). GDN (Deng & Hooi, 2021) on WADI-112 is not re-trained since the authors provided the trained checkpoint of their official model.

### 4.2. Model performance overview

Table 2 outlines the model performance on the three multivariate benchmark datasets, SWaT, WADI, and SMD.

First, it is evident that all methods have higher scores on the predominantly used point-adjusted $F1_{PA}$ metric including the *random prediction* which performs better in almost all comparisons. This artificial advantage created by point-adjust is not present on the pure F1 score protocols which do not favour noisy random predictions. On both standard point-wise $F1$ and range-wise $F1_T$ metrics, the simple baselines such as PCA reconstruction error performs better on all datasets while other baselines such as 1-NN distance and L2-norm are often very close to the best performing methods. Furthermore, the NN-baselines in most cases outperform the more complex SOTA deep models which are build using these as basic building blocks. This is a strong evidence that the complicated solutions introduced to solve the TAD task do not provide a benefit compared to such simple baselines. Finally, one can notice the interplay between the point-wise and range-wise metrics. In datasets like SWAT, where there is a small number of long anomaly intervals, the $F1$ score is much higher than the $F1_T$ score, on noisy datasets with more consistent anomaly lengths, like WADI_127, $F1_T$ is tendentially higher, while on cleaner datasets with frequent short anomalies, like univariate UCR datasets, the two scores are comparable.

Table 3 provides a comparison on univariate UCR datasets with our simple baselines. Here we include two representative univariate TAD methods, a highly effective classic method Local Outlier Factor (LOF) (Breunig et al., 2000) and a more recent SOTA method Merlin (Nakamura et al., 2020). As shown in Figure 2 the normal periodical phase-

| | | UCR/IB-16 | | | UCR/IB-17 | | | UCR/IB-18 | | | UCR/IB-19 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $F1_{PA}$ | $F1$ | $F1_T$ | $F1_{PA}$ | $F1$ | $F1_T$ | $F1_{PA}$ | $F1$ | $F1_T$ | $F1_{PA}$ | $F1$ | $F1_T$ |
| | LOF (Breunig et al., 2000) | 0.878 | 0.476 | 0.476 | 1.000 | 0.959 | 0.955 | 1.000 | 0.915 | 0.911 | 1.000 | 0.857 | 0.857 |
| | MERLIN (Nakamura et al., 2020) | 1.000 | **0.846** | **0.846** | 1.000 | **0.987** | **0.987** | 1.000 | 0.795 | 0.795 | 1.000 | 0.870 | 0.870 |
| **Simple baselines** | Random | 0.151 | 0.005 | 0.030 | 0.941 | 0.041 | 0.116 | 0.887 | 0.039 | 0.039 | 0.488 | 0.030 | 0.030 |
| | Sensor range deviation | 0.000 | 0.003 | 0.000 | 0.902 | 0.085 | 0.094 | 0.000 | 0.038 | 0.038 | 0.000 | 0.004 | 0.004 |
| | L2-norm | 0.014 | 0.011 | 0.021 | 0.276 | 0.058 | 0.164 | 0.241 | 0.061 | 0.061 | 0.028 | 0.017 | 0.017 |
| | 1-NN distance | 0.828 | 0.786 | 0.786 | 1.000 | 0.973 | 0.969 | 1.000 | 0.889 | 0.889 | 1.000 | 0.870 | 0.870 |
| | PCA Error | 0.889 | 0.750 | 0.750 | 1.000 | 0.974 | 0.974 | 1.000 | **0.990** | **0.990** | 1.000 | **1.000** | **1.000** |

*Table 3.* Comparison of simple baselines on four univariate UCR/InternalBleeding datasets.

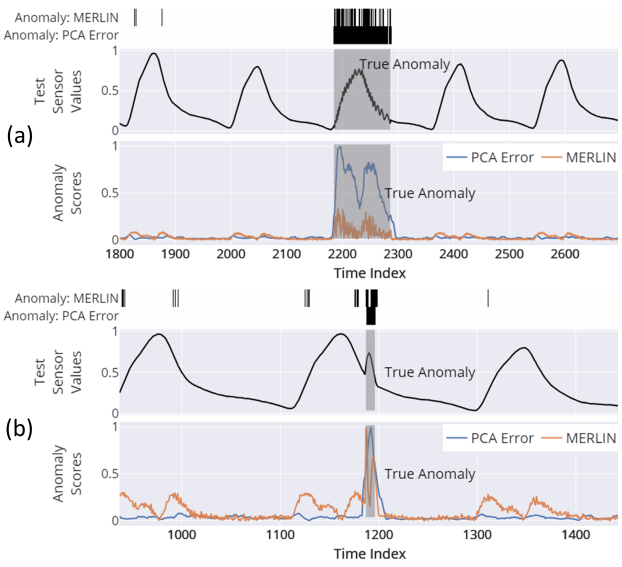



*Figure 2.* Visual comparison: The gray shaded areas denote the ground truth anomalies. (a) UCR/IB-18 dataset with a series of sine waves added as anomaly. (b) UCR/IB-19 dataset with random numbers added as anomaly.

shift and magnitude changes, which are considered normal in the light of physiology, are misclassified as anomalies by such methods in contrast to the simple PCA-Error baseline.

### 4.3. Analysis of the deep models learned function

The consistently better results of the simple methods raises the question of what type of functions are learned by the more complicated deep learning models. To investigate this, we try to approximate the behavior of the most prominent of the deep learning models by linear functions. We achieve this by performing a simple form of distillation. Given a deep learning model $M_\theta$ trained on the training data $\mathbf{X}$, we compute its predictions $M(\mathbf{X}) \subset \mathbb{R}^F$ and then train a linear model $L$ on the data/target tuple $(\mathbf{X}, M(\mathbf{X}))$ using a mean squared error (MSE) loss. The linear model in this case is simply a 1-layer perceptron. Upon evaluating both $M$ and $L$ on the test set on the anomaly detection task, we observed that their scores are very close and they exhibited high agreement on their predictions. Table 4 depicts

| Methods | SWaT | | WADI_112 | |
|---|---|---|---|---|
| | Orig | Line | Orig | Line |
| Single block MLPMixer | 0.780 | 0.770 | 0.497 | 0.500 |
| Single Transformer block | 0.787 | 0.772 | 0.534 | 0.521 |
| 1-Layer GCN-LSTM | 0.829 | 0.794 | 0.596 | 0.587 |
| TranAD (Tuli et al., 2022) | 0.799 | 0.800 | 0.511 | 0.572 |
| GDN (Deng & Hooi, 2021) | 0.810 | 0.808 | 0.571 | 0.543 |

*Table 4.* Linear approximation of complex models on two datasets. **Orig**: original model **Line**: linear approximated mode. Performance is reported on the standard point-wise $F1$ score.

this with the linear model $L$ marked as 'Line' and the corresponding deep learning model $M$ marked as 'Orig'. The performance of distilled linear version of the complex models suggests that even though the learned functions may be complex and may improve forecasting, their ability to distinguish anomalies can still be effectively captured by linearizing them.

### 4.4. Ablation: Impact of normalization

Anomaly detection methods for multivariate datasets often employ normalization and smoothing techniques to address abrupt changes in prediction scores that are not accurately predicted. However, the choice of normalization method before thresholding can impact performance on different datasets. In Table 5, we compared the performance with and without normalization. We consider two normalization methods, mean-standard deviation and median-IQR, on two datasets. Our analysis shows that median-IQR normalization, which is also utilized in the GDN (Deng & Hooi, 2021) method, improves performance on noisier datasets such as WADI. In Table 2, we have presented the best performance achieved by each method, including our baselines and considered state-of-the-art models, using either none or one of these normalisation, whichever is applicable.

### 4.5. Ablation: PCA Error projection dimension

On all the multivariate datasets with more than 50 sensors (i.e., SWaT and WADI) our PCA Error baseline approach utilized the first 30 eigenvectors for the PCA projection. In Figure 3, we present the performance as a function of vary-

| | SWAT | | | | | | WADI-112 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | None | | Mean-STD | | Median-IQR | | None | | Mean-STD | | Median-IQR | |
| | F1 | AUPRC | F1 | AUPRC | F1 | AUPRC | F1 | AUPRC | F1 | AUPRC | F1 | AUPRC |
| GDN (Deng & Hooi, 2021) | 0.767 | 0.724 | 0.685 | 0.473 | 0.810 | 0.762 | 0.549 | 0.492 | 0.456 | 0.353 | 0.571 | 0.519 |
| TranAD (Tuli et al., 2022) | 0.769 | 0.708 | 0.742 | 0.638 | 0.799 | 0.764 | 0.511 | 0.529 | 0.448 | 0.312 | 0.509 | 0.554 |
| PCA Error | 0.802 | 0.725 | 0.833 | 0.744 | 0.756 | 0.721 | 0.600 | 0.591 | 0.513 | 0.351 | 0.654 | 0.570 |
| 1-Layer GCN-LSTM | 0.770 | 0.715 | 0.775 | 0.660 | 0.829 | 0.792 | 0.592 | 0.520 | 0.520 | 0.411 | 0.596 | 0.535 |

*Table 5.* Impact of normalization on scores. Normalisation of prediction scores before thresholding impacts performance. Performance is reported on the point-wise $F1$ and $AUPRC$ score.

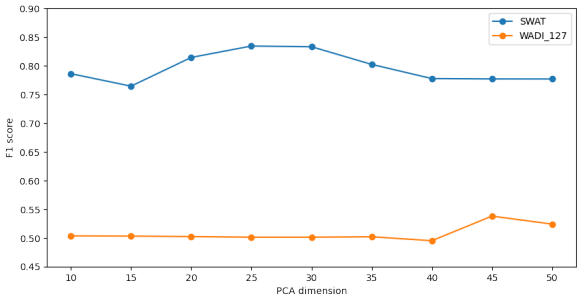



*Figure 3.* Point-wise F1 score as a function of the PCA dimension for the PCA Error method, evaluated on the SWAT and WADI_127 datasets.

ing PCA projection dimensions. It is observed that higher projection dimensions may be more beneficial for WADI (127-dimensional) compared to SWaT (51-dimensional). However, the optimal projection dimension should be determined using a validation set as it may impact performance. Unlike more sophisticated techniques with several hyperparameters specifically configured for each dataset, the baseline approach of using PCA with a fixed number of eigenvectors is relatively simple and easily tunable.

## 5. Quo vadis

As we have demonstrated, a plethora of deep learning approaches introduced to solve the task of TAD were outperformed by simple neural networks and linear baselines. Furthermore, when distilling some of those methods to linear models, their performance remained almost unchanged. There could be several causes for this issue for example the over-fitting on the normal data or the existence of too high aleatoric uncertainty which makes it hard to separate the difficult anomalies from normal sections. In any case, the main takeaway is that those methods, though potentially useful for other time-series tasks such as forecasting, do not bring much additional value for the task of TAD and their complexity is definitely not justified. What is even more worrisome, is that they managed to create up to now an illusion of progress due to the use of a flawed evaluation protocol, inadequate metrics and the lack or low quality of benchmarking with simpler methods.

We cannot stress enough the fact that almost all the recent deep-learning based methods use the point-adjust post-processing step often *without clearly stating* this. Under this evaluation these models implicitly optimize for near random predictions where their high performance is used as evidence of their proposed model's utility. An example of this trend presented at recent leading Machine Learning venues is (Xu et al., 2022; Li et al., 2023; Zhou et al., 2023). Another common malpractice, is the use of mismatched evaluation metrics in tables i.e., applying point-adjust and directly comparing their results to other methods which were scored without it. Similar issues are observed in dataset discrepancies like the introduction of new versions of a dataset which use a subset of the sensors and result in higher scores.

Aside from exposing the limitations of these methods, we provide a comprehensive set of simple benchmarks which can help re-start investigations in TAD starting on a solid baseline. We think that those methods will pinpoint which anomalies are easy to detect and which ones are the challenging ones that should be detected if any progress is to be made. This is further reinforced by the fact that there seem to be a high agreement between detected and undetected anomalies between all methods investigated. We provide an analysis of this agreement in the appendix section A.1. This agreement leads us to believe that the current datasets used in TAD are, in some sense, simultaneously too hard and too easy. The fact that so many complex deep learning architectures have been developed to tackle the hard anomalies in those datasets, but failed, is unsatisfactory, but maybe not unexpected. More comprehensive datasets with a spread spectrum of difficulty in anomalies could provide an incremental improvement path and means of properly comparing methods.

Furthermore, we believe that evaluation using both point-wise and range-wise methods will help better compare methods and identify their strengths and weaknesses.

We hope our work will help improve the research efforts on TAD by triggering focus on the introduction of new and richer datasets, increasing awareness of limitations of current evaluation protocols, and encouraging caution in the premature adoption of complex tools for the task.

## Impact Statement

"This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here."

## References


Ahmed, C. M., Palleti, V. R., and Mathur, A. P. 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, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349758. doi: 10.1145/3055366.3055375.

Angiulli, F. and Pizzuti, C. Fast outlier detection in high dimensional spaces. In Elomaa, T., Mannila, H., and Toivonen, H. (eds.), *Principles of Data Mining and Knowledge Discovery*, pp. 15–27, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45681-0.

Audibert, J., Michiardi, P., Guyard, F., Marti, S., and Zuluaga, M. A. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 3395–3404, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403392. URL https://doi.org/10.1145/3394486.3403392.

Audibert, J., Michiardi, P., Guyard, F., Marti, S., and Zuluaga, M. A. Do deep neural networks contribute to multivariate time series anomaly detection? *Pattern Recogn.*, 132(C), dec 2022. ISSN 0031-3203. doi: 10.1016/j.patcog.2022.108945. URL https://doi.org/10.1016/j.patcog.2022.108945.

Baldi, P. and Hornik, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989. ISSN 0893-6080.

Bandaragoda, T., Ting, K., Albrecht, D., Liu, F. T., Zhu, Y., and Wells, J. Isolation-based anomaly detection using nearest-neighbor ensembles: inne. *Computational Intelligence*, 34, 01 2018. doi: 10.1111/coin.12156.

Blázquez-García, A., Conde, A., Mori, U., and Lozano, J. A. A review on outlier/anomaly detection in time series data. *ACM Comput. Surv.*, 54(3), apr 2021. ISSN 0360-0300. doi: 10.1145/3444690. URL https://doi.org/10.1145/3444690.

Boniol, P., Palpanas, T., Meftah, M., and Remy, E. Graphan: Graph-based subsequence anomaly detection. *Proc. VLDB Endow.*, 13(12):2941–2944, aug 2020.

Boniol, P., Paparrizos, J., Palpanas, T., and Franklin, M. J. Sand: Streaming subsequence anomaly detection. *Proc. VLDB Endow.*, 14(10):1717–1729, jun 2021. ISSN 2150-8097. doi: 10.14778/3467861.3467863. URL https://doi.org/10.14778/3467861.3467863.

Bourlard, H. and Kamp, Y. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59:291–4, 02 1988. doi: 10.1007/BF00332918.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

Chen, Z., Chen, D., Zhang, X., Yuan, Z., and Cheng, X. Learning graph structures with transformer for multivariate time series anomaly detection in iot. *IEEE Internet of Things Journal*, pp. 1–1, 2021. doi: 10.1109/JIOT.2021.3100509.

Deng, A. and Hooi, B. Graph neural network-based anomaly detection in multivariate time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4027–4035, May 2021. doi: 10.1609/aaai.v35i5.16523. URL https://ojs.aaai.org/index.php/AAAI/article/view/16523.

Eamonn Keogh. Problems with time series anomaly detection. Personal communication, Distinguished Professor and Ross Family Chair, University of California Riverside, USA, 7 May 2024. URL https://drive.google.com/file/d/1DpAK92HNAZBjDDFdelFh-c7P4C4q_xaQ/view?usp=share_link.

Feng, C. and Tian, P. Time series anomaly detection for cyber-physical systems via neural system identification and bayesian filtering. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pp. 2858–2867, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467137. URL https://doi.org/10.1145/3447548.3467137.

Guillame-Bert, M. and Dubrawski, A. Classification of time sequences using graphs of temporal constraints. *Journal of Machine Learning Research*, 18(121):1–34, 2017. URL http://jmlr.org/papers/v18/15-403.html.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Soderstrom, T. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 387–395, New York, NY, USA, 2018a. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219845. URL https://doi.org/10.1145/3219819.3219845.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Soderstrom, T. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 387–395, New York, NY, USA, 2018b. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219845. URL https://doi.org/10.1145/3219819.3219845.

Kanarachos, S., Christopoulos, S.-R. G., Chroneos, A., and Fitzpatrick, M. E. Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and hilbert transform. *Expert Syst. Appl.*, 85(C):292–304, nov 2017.

Kim, S., Choi, K., Choi, H.-S., Lee, B., and Yoon, S. Towards a rigorous evaluation of time-series anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7194–7201, 2022. doi: 10.1609/aaai.v36i7.20680. URL https://ojs.aaai.org/index.php/AAAI/article/view/20680.

Li, D., Chen, D., Jin, B., Shi, L., Goh, J., and Ng, S.-K. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. pp. 703–716, 2019. doi: 10.1007/978-3-030-30490-4_56. URL https://doi.org/10.1007%2F978-3-030-30490-4_56.

Li, Y., Chen, W., Chen, B., Wang, D., Tian, L., and Zhou, M. Prototype-oriented unsupervised anomaly detection for multivariate time series. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008. doi: 10.1109/ICDM.2008.17.

Mathur, A. P. and Tippenhauer, N. O. 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. 31–36, 2016. doi: 10.1109/CySWater.2016.7469060.

Nakamura, T., Imamura, M., Mercer, R., and Keogh, E. Merlin: Parameter-free discovery of arbitrary length anomalies in massive time series archives. In *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 1190–1195, 2020. doi: 10.1109/ICDM50108.2020.00147.

Paparrizos, J., Kang, Y., Boniol, P., Tsay, R. S., Palpanas, T., and Franklin, M. J. Tsb-uad: an end-to-end benchmark suite for univariate time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(8):1697–1711, 2022.

Park, D., Hoshi, Y., and Kemp, C. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, PP, 11 2017. doi: 10.1109/LRA.2018.2801475.

Patcha, A. and Park, J.-M. J. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51:3448–3470, 08 2007. doi: 10.1016/j.comnet.2007.02.001.

Salem, O., Guerassimov, A., Mehaoua, A., Marcus, A., and Furht, B. Anomaly detection in medical wireless sensor networks using svm and linear regression models. *Int. J. E-Health Med. Commun.*, 5(1):20–45, jan 2014.

Schmidl, S., Wenig, P., and Papenbrock, T. Anomaly detection in time series: A comprehensive evaluation. *Proc. VLDB Endow.*, 15(9):1779–1797, may 2022. ISSN 2150-8097. doi: 10.14778/3538598.3538602. URL https://doi.org/10.14778/3538598.3538602.

Shen, L., Li, Z., and Kwok, J. Timeseries anomaly detection using temporal hierarchical one-class network. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13016–13026. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/97e401a02082021fd24957f852e0e475-Paper.pdf.

Shyu, M.-L., Chen, S.-C., Sarinnapakorn, K., and Chang, L. *Principal Component-based Anomaly Detection Scheme*, pp. 311–329. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., and Pei, D. 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*, KDD '19,

pp. 2828–2837, New York, NY, USA, 2019a. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330672. URL https://doi.org/10.1145/3292500.3330672.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., and Pei, D. 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*, KDD '19, pp. 2828–2837, New York, NY, USA, 2019b. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330672. URL https://doi.org/10.1145/3292500.3330672.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., and Pei, D. 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*, KDD '19, pp. 2828–2837, New York, NY, USA, 2019c. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330672. URL https://doi.org/10.1145/3292500.3330672.

Tatbul, N., Lee, T. J., Zdonik, S., Alam, M., and Gottschlich, J. Precision and recall for time series. *Advances in neural information processing systems*, 31, 2018.

Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A. P., Keysers, D., Uszkoreit, J., Lucic, M., and Dosovitskiy, A. MLP-mixer: An all-MLP architecture for vision. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=EI2KOXKdnP.

Tran, L., Mun, M. Y., and Shahabi, C. Real-time distance-based outlier detection in data streams. *Proc. VLDB Endow.*, 14(2):141–153, oct 2020. ISSN 2150-8097. doi: 10.14778/3425879.3425885. URL https://doi.org/10.14778/3425879.3425885.

Tuli, S., Casale, G., and Jennings, N. R. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. *Proceedings of VLDB*, 15(6): 1201–1214, 2022.

Wagner, D., Michels, T., Schulz, F. C., Nair, A., Rudolph, M., and Kloft, M. TimeseAD: Benchmarking deep multivariate time-series anomaly detection. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=iMmsCI0JsS.

Wu, R. and Keogh, E. J. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress (extended abstract). In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 1479–1480, 2022. doi: 10.1109/ICDE53745.2022.00116.

Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., Chen, J., Wang, Z., and Qiao, H. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pp. 187–196, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN 9781450356398. doi: 10.1145/3178876.3185996. URL https://doi.org/10.1145/3178876.3185996.

Xu, J., Wu, H., Wang, J., and Long, M. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LzQQ89U1qm_.

Yaacob, A. H., Tan, I. K., Chien, S. F., and Tan, H. K. Arima based network anomaly detection. In *2010 Second International Conference on Communication Software and Networks*, pp. 205–209, 2010. doi: 10.1109/ICCSN.2010.55.

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., and Chawla, N. V. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1409–1416, Jul. 2019. doi: 10.1609/aaai.v33i01.33011409. URL https://ojs.aaai.org/index.php/AAAI/article/view/3942.

Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., and Zhang, Q. Multivariate time-series anomaly detection via graph attention network. In *IEEE International Conference on Data Mining (ICDM)*, pp. 841–850, 2020.

Zhou, T., Niu, P., Sun, L., Jin, R., et al. One fits all: Power general time series analysis by pretrained lm. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 43322–43355, 2023.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

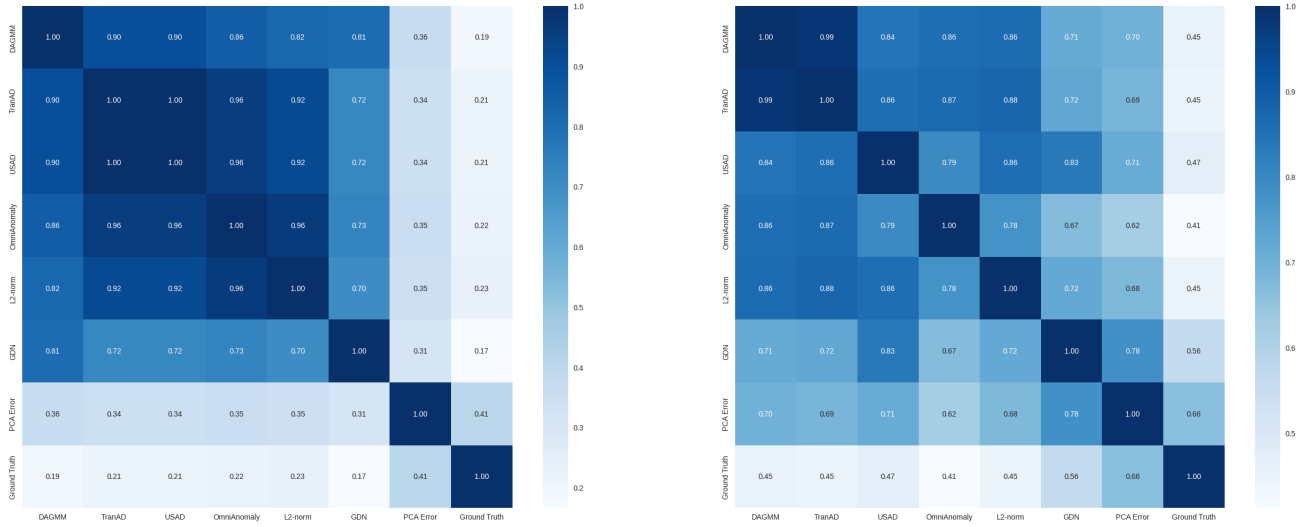
# A. Appendix

In the following appendix, we present several analyses and ablation studies related to the results discussed in the main paper. It is structured as follows:

1. *Analysis*: We analyze the agreement on the detected anomalies between the different models (Figures 4a and 4b).
2. *Additional evaluations/ablations*: Several studies are presented related to the evaluation of the model performances:
   - *Ablation window size for Univariate data*: We show the impact of the sliding window size on the performance of our simple baselines on univariate data (Figure 5).
   - *NN-baselines: reconstruction vs forecasting mode*: We show the performance of our neural network baselines when trained in reconstruction and forecasting mode (Table 6).
   - *Detailed performance comparison*: At the end, we include detailed tables (Table 9, Table 10, Table 11, Table 12) with performance comparison of all methods reporting their F1, precision, recall and AUPRC under both standard point-wise and time-series range-wise metrics.
3. *Performance of our simple baselines on SMAP and MSL datsets*: We include a comparison of our simple baseline methods and various SOTA methods on the additional multivariate SMAP and MSL datasets (Table 8).

## A.1. Analysis of model agreement on the detected anomalies

We have noticed a very high agreement on the anomalies detected by the different methods. Those agreements are especially pronounced between the SOTA deep learning methods. In order to quantify them, we compute a score similar to mAP in object detection which measures the agreement between two different predictions restricted to the ground truth anomaly intervals. The score is defined as follows:




(a) SWAT agreement matrix between methods expressed as the IOU of the sets of interval indices averaged over the hit ratio thresholds in $[0.2 : 0.95 : 0.05]$.

(b) WADI_112 agreement matrix between methods expressed as the IOU of the sets of interval indices averaged over the hit ratio thresholds in $[0.2 : 0.95 : 0.05]$.

*Figure 4.* Analysis of model agreement on the detected anomalies

Assume $A = \{[a_1, b_1], \dots [a_K, b_K]\}$ are the $K$ ground truth anomaly intervals, defined by their start and end timestamp indices as integer intervals. Thus for the interval on index $s$, $\hat{y}_i = 1$ for all $t \in [a_s, b_s]$. For an interval $[a_s, b_s]$ and a prediction $\tilde{y}$, the hit ratio is the ratio $\frac{|\{t \in [a_s, b_s] : \tilde{y}_t = 1\}|}{|[a_s, b_s]|}$ of the timestamps with a positive prediction in $[a_s, b_s]$ to the total number of timestamps in $[a_s, b_s]$. For a given prediction $\tilde{y}$ and a hit ratio threshold $r$, the detected anomaly intervals is the index list $H_{\tilde{y}} = \{i_1, \dots i_L\} \subseteq [1, L]$ of intervals for which the prediction has hit ratio above $r$. For two different predictions $\tilde{y}^1$ and $\tilde{y}^2$, the agreement between them on a given hit ratio threshold $r$ is defined as the intersection over union

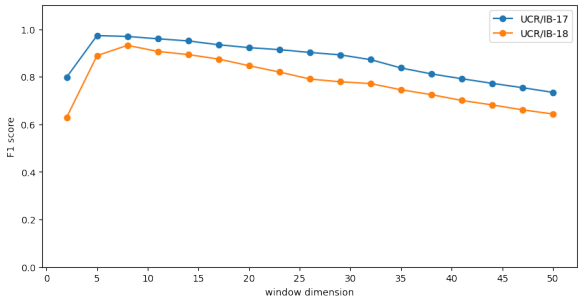



*Figure 5.* Impact of sliding window size to generate univariate data representation on the two UCR dataset traces UCR/IB-17 and UCR/IB-18.

| Method | Reconstruction | | | | | | Forecasting | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SWAT | | WADI 127 | | WADI 112 | | SWAT | | WADI 127 | | WADI 112 | |
| | F1 | AUPRC | F1 | AUPRC | F1 | AUPRC | F1 | AUPRC | F1 | AUPRC | F1 | AUPRC |
| 1-Layer MLP | 0.771 | 0.707 | 0.161 | 0.139 | 0.580 | 0.517 | 0.771 | 0.797 | 0.267 | 0.193 | 0.501 | 0.437 |
| Single block MLPmixer | 0.770 | 0.706 | 0.289 | 0.210 | 0.530 | 0.475 | 0.779 | 0.791 | 0.274 | 0.217 | 0.496 | 0.426 |
| Single Transformer Block | 0.770 | 0.702 | 0.448 | 0.410 | 0.527 | 0.798 | 0.795 | 0.874 | 0.319 | 0.662 | 0.538 | 0.756 |
| 1-Layer GCN-LSTM | 0.770 | 0.714 | 0.498 | 0.451 | 0.577 | 0.486 | 0.829 | 0.792 | 0.439 | 0.367 | 0.596 | 0.535 |

*Table 6.* NN-Baselines: Reconstruction vs Forecasting.

(IOU) of the index sets $H_{\tilde{y}^1}, H_{\tilde{y}^2}$ for $r$. Finally, the average agreement between two predictions $\tilde{y}^1$ and $\tilde{y}^2$ is the mean of their agreements over all the thresholds from 0.2 to 0.95 with step 0.05.

In Figures 4a and 4b the matrices of the agreements between all models and the ground truth are displayed for the SWAT and WADI-112 datasets. In both cases, the agreement between different models is much higher compared to the agreement to the ground truth, indicating that the models learn to recognize similar anomalies. Only the GDN model and even more the PCA Error baseline seem to have a comparably higher agreement with the ground truth.

## A.2. Additional evaluations/ablations

### A.2.1. Ablation window size for Univariate data

As outlined in section 3.2 of the main paper, we created an effective univariate data representation by concatenating past observations with the current timestamp using a sliding window approach. We discovered that this basic representation yielded effective results with a window size of $w = 4$ leading to a 5-dimensional representation space. Figure 5 displays the performance impact based on the window size. This plot illustrates that a smaller window over 4-5 past observations is a reasonable choice for the UCR datasets, while larger window dimensions do not add any further advantage. We opted to use our simple 1-NN distance approach and varied the window sizes to avoid manipulating any other parameters.

### A.2.2. NN-baselines: reconstruction vs forecasting mode

In our main paper, we demonstrated the effectiveness of our simple neural network baselines when trained in forecasting mode, which is in line with most state-of-the-art deep learning models we compared with. During training, the output before the final target dense regression layer has a shape of (batch-size, sequence, embedding-dim). In forecasting mode, we use a 1-D global average pooling to project it to (batch-size, 1, embedding-dim). However, we can skip the average-pooling operation and train these models in a reconstruction (auto-encoding) fashion. For completeness, we present their performance in reconstruction mode in Table 6. Our results show that the performance of these models in reconstruction mode is comparable to that in forecasting mode, particularly considering the impact of random seed between training runs. Therefore, there does not appear to be any significant advantage in training these models in forecasting mode, at least for the datasets we considered.

### A.2.3. DETAILED PERFORMANCE COMPARISON

Finally, we provide tables which contain the detailed scores of all models in terms of precision, recall, F1-score and area under the precision-recall curve (AUPRC). For the multivariate time series datasets, Table 9 shows the evaluation under point-wise metrics; Table 10 shows the evaluation under time series range-wise metrics (Wagner et al., 2023). Similarly, for the univariate datasets, Table 11 evaluates under point-wise and Table 12 provides the performance under range-wise metrics.

## A.3. Performance of our simple baselines on SMAP and MSL datasets

Soil Moisture Active Passive (SMAP) and Mars Science Laboratory (MSL) datasets, collected from a spacecraft of NASA (Hundman et al., 2018b), are another two widely utilized benchmark datasets in the literature. The SMAP dataset contains information on soil samples and telemetry of the Mars rover; the MSL dataset comes from the actuator and sensor data for the Mars rover itself. Although these benchmark datasets are widely used in the literature, their quality and validity suffer from several pitfalls, such as triviality, mislabeling, and unrealistic density of anomaly (see Wu & Keogh (2022) for details). The statistics profile of each dataset is listed in Table 7. Since each dataset contains traces with various lengths in both the training and test sets, we report the average length of traces and the average number of anomalies among all traces per dataset. We also report the total number of data points and anomalies per dataset for the clarity of comparison in the literature.

| Dataset | No. Sensors (Traces) | Avg. Train (Total) | Avg. Test (Total) | Avg. Anomalies (%) | Total Anomalies (%) |
|---|---|---|---|---|---|
| MSL | 55 (27) | 2159 (58317) | 2730 (73729) | 286 (11.97%) | 7730 (10.48%) |
| SMAP | 25 (54 ) | 2555 (138004) | 8070 (435826) | 1034 (12.40%) | 55854 (12.82%) |

*Table 7.* The statistical profile of the datasets: MSL and SMAP.

Table 8 summarizes the point-adjust F1 and standard F1 scores of simple baseline models and the published performance of the SOTA models. The performance of each proposed simple baseline model is averaged over all traces per dataset. The results of SOTA methods are taken from Kim et al.2022 in which only the best F1 scores are reported per method. The simple baselines, namely PCA-error and 1-NN distance, yield the best and second-best performance on both datasets, respectively.

| | Method | Datasets | | | |
|---|---|---|---|---|---|
| | | MSL | | SMAP | |
| | | $F1_{PA}$ | $F1$ | $F1_{PA}$ | $F1$ |
| **Simple baselines** | Random | **0.931** | 0.190 | **0.961** | 0.227 |
| | Sensor range deviation | 0.441 | 0.328 | 0.389 | 0.273 |
| | L2-norm | 0.854 | 0.395 | 0.745 | 0.351 |
| | 1-NN distance | 0.912 | <u>0.404</u> | <u>0.818</u> | <u>0.352</u> |
| | PCA Error | 0.843 | **0.426** | 0.811 | **0.387** |
| **SOTA Methods** | DAGMM (Zong et al., 2018) | 0.701 | 0.199 | 0.712 | 0.333 |
| | OmniAnomaly (Su et al., 2019a) | 0.899 | 0.207 | 0.805 | 0.227 |
| | USAD (Audibert et al., 2020) | <u>0.927</u> | 0.211 | <u>0.818</u> | 0.228 |
| | GDN (Deng & Hooi, 2021) | 0.903 | 0.217 | 0.708 | 0.252 |

*Table 8.* Simple baselines outperform the SOTA deep-learning models on MSL and SMAP datasets. SOTA model performance is taken from Kim et al. (2022). Bold: the best performance; underline: the second-best performance.

| Method | Datasets | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SWAT | | | | WADI 127 | | | | WADI 112 | | | | SMD | | | |
| | $F1$ | $P$ | $R$ | $AUPRC$ | $F1$ | $P$ | $R$ | $AUPRC$ | $F1$ | $P$ | $R$ | $AUPRC$ | $F1$ | $P$ | $R$ | $AUPRC$ |
| MERLIN (Nakamura et al., 2020) | 0.217 | 0.122 | 1.000 | 0.116 | 0.335 | 0.305 | 0.371 | 0.217 | 0.473 | 0.710 | 0.355 | 0.412 | 0.384 | 0.474 | 0.374 | 0.338 |
| DAGMM (Zong et al., 2018) | 0.770 | 0.991 | 0.630 | 0.727 | 0.279 | 0.993 | 0.162 | 0.207 | 0.520 | 0.932 | 0.361 | 0.469 | 0.435 | 0.564 | 0.497 | 0.370 |
| OmniAnomaly (Su et al., 2019b) | 0.773 | 0.990 | 0.634 | 0.736 | 0.281 | 1.000 | 0.163 | 0.212 | 0.441 | 0.607 | 0.346 | 0.441 | 0.415 | 0.566 | 0.464 | 0.360 |
| USAD (Audibert et al., 2020) | 0.772 | 0.988 | 0.634 | 0.730 | 0.279 | 0.993 | 0.162 | 0.207 | 0.535 | 0.744 | 0.417 | 0.483 | 0.426 | 0.546 | 0.474 | 0.364 |
| GDN (Deng & Hooi, 2021) | 0.810 | 0.987 | 0.686 | 0.762 | 0.347 | 0.643 | 0.237 | 0.304 | 0.571 | 0.727 | 0.470 | 0.519 | 0.526 | 0.597 | 0.565 | 0.457 |
| TranAD (Tuli et al., 2022) | 0.800 | 0.990 | 0.671 | 0.759 | 0.340 | 0.293 | 0.404 | 0.215 | 0.511 | 0.795 | 0.377 | 0.529 | 0.457 | 0.579 | 0.481 | 0.387 |
| AnomalyTransformer (Xu et al., 2022) | 0.765 | 0.943 | 0.643 | 0.712 | 0.209 | 0.122 | 0.743 | 0.188 | 0.543 | 0.576 | 0.513 | 0.427 | 0.426 | 0.419 | 0.528 | 0.313 |
| Random | 0.218 | 0.122 | 0.997 | 0.121 | 0.101 | 0.053 | 0.958 | 0.054 | 0.101 | 0.053 | 0.992 | 0.053 | 0.080 | 0.044 | 0.696 | 0.043 |
| Sensor Range Deviation | 0.231 | 0.131 | 0.979 | 0.556 | 0.101 | 0.053 | 1.000 | 0.317 | 0.465 | 0.567 | 0.394 | 0.497 | 0.132 | 0.110 | 0.682 | 0.321 |
| L2-norm | 0.782 | 0.985 | 0.648 | 0.715 | 0.281 | 1.000 | 0.163 | 0.210 | 0.513 | 0.887 | 0.361 | 0.474 | 0.404 | 0.569 | 0.455 | 0.343 |
| 1-NN distance | 0.782 | 0.984 | 0.649 | 0.726 | 0.281 | 1.000 | 0.163 | 0.211 | 0.568 | 0.779 | 0.447 | 0.501 | 0.463 | 0.626 | 0.458 | 0.389 |
| PCA Error | 0.833 | 0.965 | 0.733 | 0.744 | 0.501 | 0.884 | 0.350 | 0.476 | 0.655 | 0.752 | 0.580 | 0.570 | 0.572 | 0.611 | 0.584 | 0.515 |
| 1-Layer MLP | 0.771 | 0.981 | 0.635 | 0.797 | 0.267 | 0.834 | 0.159 | 0.193 | 0.502 | 0.880 | 0.351 | 0.437 | 0.514 | 0.598 | 0.574 | 0.458 |
| Single block MLPMixer | 0.780 | 0.854 | 0.718 | 0.791 | 0.275 | 0.862 | 0.163 | 0.218 | 0.497 | 0.822 | 0.356 | 0.426 | 0.512 | 0.608 | 0.554 | 0.458 |
| Single Transformer block | 0.787 | 0.868 | 0.720 | 0.821 | 0.289 | 0.908 | 0.172 | 0.255 | 0.534 | 0.735 | 0.419 | 0.453 | 0.489 | 0.589 | 0.536 | 0.422 |
| 1-Layer GCN-LSTM | 0.829 | 0.982 | 0.718 | 0.793 | 0.439 | 0.744 | 0.311 | 0.367 | 0.596 | 0.742 | 0.498 | 0.535 | 0.550 | 0.627 | 0.599 | 0.478 |

*Table 9.* Experimental results for SWaT, WADI, and SMD datasets evaluated under the standard point-wise metric.

| Method | Datasets | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SWAT | | | | WADI 127 | | | | WADI 112 | | | | SMD | | | |
| | $F1$ | $P$ | $R$ | $AUPRC$ | $F1$ | $P$ | $R$ | $AUPRC$ | $F1$ | $P$ | $R$ | $AUPRC$ | $F1$ | $P$ | $R$ | $AUPRC$ |
| MERLIN (Nakamura et al., 2020) | 0.286 | 0.521 | 0.197 | 0.180 | 0.354 | 0.308 | 0.416 | 0.247 | 0.503 | 0.748 | 0.379 | 0.466 | 0.473 | 0.641 | 0.407 | 0.406 |
| DAGMM (Zong et al., 2018) | 0.402 | 0.646 | 0.292 | 0.403 | 0.406 | 0.993 | 0.255 | 0.295 | 0.609 | 0.938 | 0.451 | 0.538 | 0.379 | 0.552 | 0.405 | 0.316 |
| OmniAnomaly (Su et al., 2019b) | 0.367 | 0.403 | 0.337 | 0.394 | 0.410 | 1.000 | 0.258 | 0.303 | 0.496 | 0.671 | 0.393 | 0.492 | 0.353 | 0.490 | 0.410 | 0.300 |
| USAD (Audibert et al., 2020) | 0.413 | 0.674 | 0.298 | 0.408 | 0.406 | 0.993 | 0.255 | 0.295 | 0.573 | 0.754 | 0.462 | 0.524 | 0.364 | 0.539 | 0.375 | 0.303 |
| GDN (Deng & Hooi, 2021) | 0.385 | 0.418 | 0.357 | 0.423 | 0.434 | 0.799 | 0.298 | 0.348 | 0.588 | 0.812 | 0.461 | 0.543 | 0.570 | 0.673 | 0.550 | 0.500 |
| TranAD (Tuli et al., 2022) | 0.425 | 0.388 | 0.471 | 0.464 | 0.353 | 0.301 | 0.425 | 0.239 | 0.589 | 0.795 | 0.468 | 0.604 | 0.390 | 0.544 | 0.399 | 0.322 |
| AnomalyTransformer (Xu et al., 2022) | 0.331 | 0.885 | 0.204 | 0.348 | 0.219 | 0.128 | 0.738 | 0.189 | 0.555 | 0.589 | 0.524 | 0.451 | 0.351 | 0.350 | 0.460 | 0.247 |
| Random | 0.217 | 0.123 | 0.951 | 0.124 | 0.106 | 0.056 | 0.919 | 0.057 | 0.106 | 0.056 | 0.952 | 0.055 | 0.080 | 0.046 | 0.707 | 0.045 |
| Sensor Range Deviation | 0.230 | 0.131 | 0.928 | 0.534 | 0.098 | 0.053 | 0.678 | 0.374 | 0.526 | 0.569 | 0.489 | 0.543 | 0.116 | 0.121 | 0.508 | 0.325 |
| L2-norm | 0.366 | 0.898 | 0.230 | 0.367 | 0.410 | 1.000 | 0.258 | 0.300 | 0.607 | 0.908 | 0.456 | 0.582 | 0.338 | 0.461 | 0.411 | 0.281 |
| 1-NN distance | 0.372 | 0.937 | 0.232 | 0.391 | 0.410 | 1.000 | 0.258 | 0.301 | 0.618 | 0.915 | 0.467 | 0.564 | 0.384 | 0.517 | 0.389 | 0.327 |
| PCA Error | 0.574 | 0.918 | 0.417 | 0.504 | 0.557 | 0.884 | 0.406 | 0.543 | 0.699 | 0.752 | 0.652 | 0.640 | 0.580 | 0.641 | 0.597 | 0.516 |
| 1-Layer MLP | 0.519 | 0.740 | 0.400 | 0.532 | 0.384 | 0.834 | 0.250 | 0.280 | 0.558 | 0.880 | 0.408 | 0.493 | 0.487 | 0.536 | 0.513 | 0.424 |
| Single block MLPMixer | 0.549 | 0.762 | 0.430 | 0.549 | 0.396 | 0.862 | 0.257 | 0.307 | 0.552 | 0.865 | 0.405 | 0.484 | 0.472 | 0.525 | 0.556 | 0.426 |
| Single Transformer block | 0.526 | 0.556 | 0.500 | 0.573 | 0.416 | 0.908 | 0.270 | 0.354 | 0.575 | 0.735 | 0.472 | 0.506 | 0.420 | 0.531 | 0.439 | 0.370 |
| 1-Layer GCN-LSTM | 0.532 | 0.914 | 0.375 | 0.532 | 0.540 | 0.745 | 0.424 | 0.468 | 0.645 | 0.742 | 0.570 | 0.599 | 0.535 | 0.591 | 0.566 | 0.462 |

*Table 10.* Experimental results for SWaT, WADI, and SMD datasets evaluated under the time-series range-wise metric.

| Method | Datasets | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | UCR/IB-16 | | | | UCR/IB-17 | | | | UCR/IB-18 | | | | UCR/IB-19 | | | |
| | $F1$ | $P$ | $R$ | $AUPRC$ | $F1$ | $P$ | $R$ | $AUPRC$ | $F1$ | $P$ | $R$ | $AUPRC$ | $F1$ | $P$ | $R$ | $AUPRC$ |
| LOF (Breunig et al., 2000) | 0.476 | 0.555 | 0.416 | 0.196 | 0.959 | 0.955 | 0.963 | 0.944 | 0.916 | 0.920 | 0.911 | 0.832 | 0.857 | 0.818 | 0.900 | 0.939 |
| MERLIN (Nakamura et al., 2020) | 0.846 | 0.786 | 0.917 | 0.871 | 0.987 | 0.982 | 0.991 | 0.979 | 0.795 | 0.986 | 0.667 | 0.724 | 0.870 | 0.769 | 1.000 | 0.945 |
| Random | 0.005 | 0.002 | 0.500 | 0.002 | 0.041 | 0.024 | 0.144 | 0.018 | 0.039 | 0.020 | 0.725 | 0.017 | 0.030 | 0.016 | 0.200 | 0.006 |
| Sensor range deviation | 0.004 | 0.002 | 1.000 | 0.001 | 0.085 | 0.200 | 0.054 | 0.136 | 0.038 | 0.020 | 1.000 | 0.010 | 0.004 | 0.002 | 1.000 | 0.001 |
| L2-norm | 0.011 | 0.005 | 1.000 | 0.003 | 0.058 | 0.030 | 0.748 | 0.024 | 0.061 | 0.032 | 0.794 | 0.026 | 0.017 | 0.008 | 1.000 | 0.005 |
| 1-NN distance | 0.786 | 0.688 | 0.917 | 0.471 | 0.973 | 0.965 | 0.982 | 0.992 | 0.889 | 0.876 | 0.902 | 0.961 | 0.870 | 0.769 | 1.000 | 0.788 |
| PCA Error | 0.750 | 0.600 | 1.000 | 0.737 | 0.974 | 0.949 | 1.000 | 0.997 | 0.990 | 0.981 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |

*Table 11.* Experimental results for four univariate UCR/InternalBleeding datasets evaluated under the standard point-wise metric.

| Method | Datasets | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | UCR/IB-16 | | | | UCR/IB-17 | | | | UCR/IB-18 | | | | UCR/IB-19 | | | |
| | $F1$ | $P$ | $R$ | $AUPRC$ | $F1$ | $P$ | $R$ | $AUPRC$ | $F1$ | $P$ | $R$ | $AUPRC$ | $F1$ | $P$ | $R$ | $AUPRC$ |
| LOF (Breunig et al., 2000) | 0.476 | 0.555 | 0.416 | 0.223 | 0.955 | 0.947 | 0.964 | 0.946 | 0.911 | 0.920 | 0.902 | 0.837 | 0.857 | 0.818 | 0.900 | 0.939 |
| MERLIN (Nakamura et al., 2020) | 0.846 | 0.786 | 0.917 | 0.872 | 0.987 | 0.982 | 0.991 | 0.981 | 0.791 | 0.986 | 0.660 | 0.763 | 0.870 | 0.769 | 1.000 | 0.941 |
| Random | 0.030 | 0.016 | 0.229 | 0.005 | 0.116 | 0.062 | 0.947 | 0.062 | 0.091 | 0.050 | 0.469 | 0.043 | 0.048 | 0.031 | 0.100 | 0.008 |
| Sensor range deviation | 0.000 | 0.000 | 1.000 | 0.001 | 0.094 | 0.353 | 0.054 | 0.212 | 0.000 | 0.000 | 1.000 | 0.010 | 0.000 | 0.000 | 1.000 | 0.001 |
| L2-norm | 0.021 | 0.010 | 1.000 | 0.006 | 0.164 | 0.092 | 0.734 | 0.076 | 0.123 | 0.067 | 0.794 | 0.054 | 0.050 | 0.026 | 1.000 | 0.016 |
| 1-NN distance | 0.786 | 0.688 | 0.917 | 0.480 | 0.969 | 0.957 | 0.982 | 0.992 | 0.902 | 0.828 | 0.990 | 0.961 | 0.870 | 0.769 | 1.000 | 0.791 |
| PCA Error | 0.750 | 0.600 | 1.000 | 0.708 | 0.974 | 0.949 | 1.000 | 0.997 | 0.990 | 0.981 | 1.000 | 0.999 | 1.000 | 1.000 | 1.000 | 1.000 |

*Table 12.* Experimental results for four univariate UCR/InternalBleeding datasets evaluated under the time-series range-wise metric.