Unsuperivesed Time-series Anomaly Detection

Group17: Weitao Li 50022319
The Hong Kong University of Science and Technology (Guangzhou)
wli741@connect.hkust-gz.edu.cn

Abstract

Anomaly detection in time series data is a fundamental task with wide-ranging applications, from fraud detection to system monitoring. In this paper, we present a comprehensive evaluation of unsupervised anomaly detection methods on both univariate and multivariate time series datasets. We propose DLinear, a lightweight linear forecasting model, as our primary method, and benchmark it against a diverse set of baseline models including reconstruction-based, prediction-based, and clustering-based approaches. To ensure fair and reliable comparison, we adopt the TSB-AD benchmark framework and employ a variety of evaluation metrics, including AUC-ROC, VUS-PR, and range-aware F1 scores. We analyze model performance, stability, and runtime efficiency across two real-world datasets: a medical dataset from the UCR archive and a high-dimensional system dataset from Exathlon. Our results show that clustering methods such as KMeansAD_U excel in univariate settings, while simple statistic model PCA and deep learning models like OmniAnomaly achieve superior results in multivariate contexts. Additionally, we highlight the trade-offs between detection accuracy and computational cost, emphasizing the need to match models with specific application requirements.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Background

As commonly defined in the literature [9], anomalies refer to data points (single points or group of points) that do not conform to some notion of normality or an expected behavior based on previously observed data. In practice, anomalies can correspond to erroneous data (e.g., broken sensors) or data of interest (e.g., anomalous behavior of the measured system) [1]. Detecting such cases is crucial for many applications [8].

Point and contextual anomalies, refer to data points deviating remarkably from the rest of the data globally or given a specific context, respectively.

1.1 Time Series Decomposition

It is possible to decompose a time series X into four components, each of which express a specific aspect of its movement [6]. The components are as follows:

- Secular trend: This is the long-term trend in the series, such as increasing, decreasing or stable. The secular trend represents the general pattern of the data over time and does not have to be linear. The change in population in a particular region over several years is an example of nonlinear growth or decay depending on various dynamic factors.
- Seasonal variations: Depending on the month, weekday, or duration, a time series may exhibit a seasonal pattern. Seasonality always occurs at a fixed frequency. For instance, a study of gas/electricity consumption shows that the consumption curve does not follow a similar pattern throughout the year. Depending on the season and the locality, the pattern is different.
- Cyclical fluctuations: A cycle is defined as an extended deviation from the underlying series defined by the secular trend and seasonal variations. Unlike seasonal effects, cyclical effects vary in onset and duration. Examples include economic cycles such as booms and recessions.
- Irregular variations: This refers to random, irregular events. It is the residual after all the other components are removed. A disaster such as an earthquake or flood can lead to irregular variations.

A time series can be mathematically described by estimating its four components separately, and each of them may deviate from the normal behaviour

1.2 Anomalies in Time Series

According to Hawkins [10], an anomaly is defined as a deviation from the general distribution of data, such as a single observation (point) or a series of observations (subsequence) that deviate significantly from the norm. Real-world data often contain noise, which may be irrelevant to the researcher [2]. The most meaningful anomalies are those that are significantly different from the norm. In time series data, trend analysis and anomaly detection are closely related but not equivalent [2]. Changes in time series datasets can also occur due to concept drift, where values and trends change gradually or abruptly over time [16].

Anomalies in time series can be classified as temporal, intermetric, or temporal-intermetric anomalies [24]. Temporal anomalies can be compared with their neighbors (local) or the entire time series (global) and take different forms depending on their behavior. Common types of temporal anomalies include:

- **Global**: Spikes in the series with extreme values compared to the rest. For example, an unusually large payment on a typical day. It can be described as:

$$|x_t - \hat{x}_t| > \text{threshold} \tag{1}$$

  where $\hat{x}_t$ is the model output. If the difference exceeds the threshold, it is recognized as an anomaly.
- **Contextual**: Deviations from neighboring time points within a certain range. These are small glitches that are normal in one context but anomalous in another. The threshold is

determined by:

$$\text{threshold} \approx \lambda \times \text{var}(X_{t-w:t}) \tag{2}$$

where var is the variance of the context.

- **Seasonal**: Unusual seasonality compared to the overall pattern. For example, abnormal customer counts in a restaurant during a week. It is measured by:

$$\text{dissimilarity}(S, \hat{S}) > \text{threshold} \tag{3}$$

where $\hat{S}$ is the expected seasonality.

- **Trend**: A permanent shift in the mean or slope of the time series. For example, a new song becoming popular and then disappearing. It is measured by:

$$\text{dissimilarity}(T, \hat{T}) > \text{threshold} \tag{4}$$

where $\hat{T}$ is the normal trend.

2 Related Work

2.1 Unsupervied Time-series Anomaly Detection

In practice, training data often contains very few labeled anomalies. Consequently, most models focus on learning the representation or features of normal data and detect anomalies by identifying deviations from these normal patterns. There are four primary learning schemes in recent deep models for anomaly detection: unsupervised, supervised, semi-supervised, and self-supervised, which are based on the availability of labeled data points.

Among these, the **unsupervised approach** is particularly noteworthy. It does not use any labels and does not distinguish between training and testing datasets. This method is highly flexible as it relies solely on the intrinsic features of the data. It is especially useful in streaming applications where labeled data is unavailable. However, evaluating anomaly detection models using unsupervised methods can be challenging due to the lack of ground truth labels. The anomaly detection problem is typically treated as an unsupervised learning problem because historical data is usually unlabeled and anomalies are unpredictable.

Supervised methods, on the other hand, require fully labeled datasets, including both normal and anomalous data points. They learn the boundaries between normal and anomalous data based on these labels and determine a threshold for classifying timestamps as anomalous. However, this method is not applicable to many real-world applications where anomalies are often unknown or improperly labeled.

Semi-supervised anomaly detection is used when the dataset only contains labeled normal data. Unlike supervised methods, it does not require fully labeled datasets. It relies on labeled normal data to define normal patterns and detects deviations as anomalies. This approach is distinct from **self-supervised learning**, where the model generates its own supervisory signal from the input data without needing explicit labels.

Deep Models for Time Series Anomaly Detection

Forecasting-Based Models

Description: Forecasting models learn the normal patterns of time series data to predict future data points. The deviation between the predicted and actual values is used as an anomaly score.
Representative Models: LSTM-based models such as LSTM-AD [15] and LSTM-PRED [5], CNN-based models like DeepAnt [17], and Transformer-based models like Anomaly Transformer [20].
Advantages: Effectively capture long-term dependencies in time series data. Suitable for large-scale and complex pattern data.
Disadvantages: May have large prediction errors for rapidly changing or unpredictable time series. Training and inference times can be long for long sequences.

Reconstruction-Based Models

Description: Reconstruction models learn to reconstruct the input time series data. The reconstruction error is used as an anomaly score.
Representative Models: Autoencoders (AE) such as EncDec-AD [3], Variational Autoencoders (VAE) like Donut [19], and Generative Adversarial Networks (GAN) like MAD-GAN [13].
Advantages: Effective in handling noisy and irregular data. Suitable for multivariate time series data.
Disadvantages: High model complexity and training difficulty. May not be sensitive enough for high-dimensional data.

Representation-Based Models

Description: Representation learning models learn low-dimensional representations of time series data to capture intrinsic features. Anomaly detection is based on deviations in these representations.
Representative Models: Transformer-based models like TS2Vec [21] and CNN-based models like TF-C [23].
Advantages: Capture multi-scale features in time series data. Suitable for large-scale datasets.
Disadvantages: Require large amounts of data for training. May not perform well on sparse data.

Research Gaps

- **High-Dimensional Data Handling:** Existing models face challenges in handling high-dimensional multivariate time series data, often resulting in decreased performance and increased computational complexity. Effective methods for dealing with high-dimensional data while maintaining model performance are needed.
- **Model Interpretability:** Although deep learning models perform well in anomaly detection, they generally lack interpretability. Enhancing the interpretability of models to provide insights into the causes of anomalies is an important direction for future research.
- **Evaluation of Unsupervised Learning:** Unsupervised learning methods are widely used in anomaly detection, but there is a lack of effective evaluation metrics and methods. Designing more reasonable evaluation metrics to accurately assess the performance of unsupervised models is an area that requires further investigation.

3 Data and Methodology

3.1 Dataset Description

In our experiment, we plan to use a dataset from UCR as a unary time series dataset and Exathlon as a multivariate time series dataset.

The UCR Time Series Classification Archive [4] is a comprehensive repository of time series datasets designed for classification tasks. It contains a wide variety of univariate time series datasets from different domains, including sensor data, medical records, and more. The datasets are carefully curated to support research in time series classification and clustering. Each dataset includes labeled time series data, making it suitable for supervised learning tasks. The UCR archive is widely used in the time series community for benchmarking and developing new algorithms.

Exathlon [11] is a benchmark dataset for explainable anomaly detection over high-dimensional time series data. It is constructed based on real data traces collected from repeated executions of large-scale stream processing jobs on an Apache Spark cluster. The dataset includes both normal and anomalous traces, with ground truth labels for the root cause intervals and extended effect intervals of the anomalies. Exathlon provides a rich and challenging testbed for developing and evaluating anomaly detection and explanation discovery algorithms. The dataset captures real-world characteristics of high-dimensional time series data, making it suitable for research in anomaly detection, explanation discovery, and their integration.

According to the project requirements, we utilize the following datasets for evaluating the proposed method:

- **Univariate Dataset:** `51_UCR_id_149_Medical_tr_3000_1st_7175.csv`
  This dataset is selected from the UCR Time Series Classification Archive and contains 3000

training samples from a medical domain. It is used to validate the model's ability to detect anomalies in univariate time series.

- **Multivariate Dataset:** `178_Exathlon_id_5_Facility_tr_12538_1st_12638.csv`
  This dataset is derived from the Exathlon benchmark and includes multivariate sensor data from a facility setting, with 12,538 training samples. It is used to assess the model's performance in detecting anomalies across multiple correlated signals.

3.2 Proposed Method

In this unsupervised anomaly detection project, we propose to use **DLinear** [22] as the core time series modeling method. The decision is grounded in the following considerations:

- **Task Alignment:** The project aims to detect anomalous behaviors from high-dimensional, noisy time series data (e.g., user activity logs). DLinear, which decomposes each input sequence into trend and residual components using a sliding window approach, is well-suited to capture local deviations typical of anomalous events.
- **Suitability for Unsupervised Reconstruction:** Many unsupervised anomaly detection methods rely on reconstruction-based paradigms. DLinear naturally fits this approach: the model learns to reconstruct the underlying trend while the residual errors can serve as a robust anomaly score.
- **Avoidance of Over-Modeling Long-Term Dependencies:** Unlike Transformer or RNN-based models that emphasize global temporal patterns, DLinear prioritizes short-term trends. This is beneficial for anomaly detection where local anomalies are often more critical than long-range dependencies.
- **Computational Efficiency:** DLinear is lightweight and fast to train, making it ideal for rapid prototyping on platforms. It integrates smoothly into our modular `model_wrapper`-based pipeline, allowing for efficient experimentation and deployment.

Based on these considerations, DLinear provides a balance between interpretability, efficiency, and detection performance, making it a strong choice for our unsupervised anomaly detection framework.

3.3 Baseline Selection Strategy

The baseline models used in this study are selected based on their reported performance in the TSB-AD benchmark [14], particularly with respect to the VUS-PR metric, which has been shown to be the most reliable evaluation measure for time-series anomaly detection. To ensure fair and comprehensive evaluation, we include representative models across different categories, including both classical and advanced methods. Specifically, we consider models from reconstruction-based, prediction-based, and clustering-based approaches.

For the **univariate datasets**, the following models are selected (with `Sub_` or `_U` indicating the use of sliding window transformation):

- **Reconstruction-based:** `Sub_PCA`, `USAD`, `OmniAnomaly`
  These methods reconstruct the input signal and use the reconstruction error to detect anomalies.
- **Prediction-based:** `LSTMAD`, `DLinear`
  These models learn to forecast the next point or sequence, and anomalies are identified via prediction deviations.
- **Clustering-based:** `KMeansAD_U`, `Sub_LOF`, `Sub_OCSVM`
  Clustering and density-based models detect anomalies by identifying points that deviate significantly from learned group structures.

For the **multivariate datasets**, we include:

- **Reconstruction-based:** `PCA`, `USAD`, `OmniAnomaly`
- **Prediction-based:** `LSTMAD`, `DLinear`

- **Clustering-based:** `KMeansAD`, `LOF`, `OCSVM`

By selecting models from each category and varying in complexity, we aim to evaluate anomaly detection performance across a broad methodological spectrum under a unified experimental setup.

3.4 Evaluation Metrics

To comprehensively assess the performance of anomaly detection models on time-series data, we adopt both point-wise and range-wise evaluation metrics, as suggested in TSB-AD [14].

For point-level anomaly detection, we adopt widely used classification-based metrics, including AUC-ROC [7] and the standard F1 score. To align with common practices in recent literature, we also report the point-adjusted F1 score (PA-F1), which considers an entire ground-truth anomaly segment as correctly detected if any of its constituent points is predicted as anomalous [12].

For range-aware evaluation, the authors of the TSB-AD benchmark [14] recommend using Volume Under the Surface (VUS) metrics to mitigate issues related to lag sensitivity and rigid binary labeling. In particular, we incorporate VUS-ROC and VUS-PR [18], and primarily rely on VUS-PR as our key evaluation metric for performance comparison under range-wise settings.

Together, these metrics enable a thorough and fair performance comparison of detection methods from both discrete and continuous anomaly perspectives.

4 Experiment Evaluation and Analysis

4.1 Hyperparameter Tuning

To ensure a fair comparison, we follow the hyperparameter tuning strategy proposed by the TSB-AD benchmark [14], where each model is optimized on a dedicated tuning dataset disjoint from the evaluation set. For all baseline models (e.g., PCA, USAD, LSTMAD), we adopt the same candidate hyperparameter ranges as reported in the TSB-AD official implementation.

For our proposed model **DLinear**, we perform hyperparameter optimization using *Bayesian Optimization*, targeting the maximization of the VUS-PR score as the objective function. The search is conducted on the same tuning datasets used in TSB-AD to maintain a consistent experimental protocol.

The parameter search space for **DLinear** includes: `window_size` $\in$ [50, 200] (integer), `lr` $\in$ [1e-4, 5e-3] (continuous), `batch_size` $\in$ [32, 128] (integer), `epochs` $\in$ [30, 200] (integer), `pred_len` $\in$ [1, 5] (integer), and `validation_size` $\in$ [0.1, 0.3] (continuous). After Bayesian optimization using the VUS-PR score as the objective, the best configuration was found to be: `window_size` = 90, `lr` = 0.005, `batch_size` = 92, `epochs` = 100, `pred_len` = 3, and `validation_size` = 0.2.

This tuned configuration is used consistently across all experiments involving DLinear on both univariate and multivariate datasets.

4.2 Model Stability Evaluation

We define **model stability** as the ability of a machine or deep learning algorithm to consistently reproduce similar results when retrained under the same conditions. While some methods, such as OC-SVM, are inherently stable due to their deterministic nature, many modern approaches—particularly neural network-based models—depend on random parameter initialization, which can introduce variance in final performance outcomes.

Ideally, a robust model should exhibit minimal sensitivity to such randomness and yield reproducible results across multiple runs. To assess this, we evaluate the stability of each model by conducting every experiment **five times** using different random seeds. We report both the **mean** and **standard deviation** of the evaluation scores (e.g., VUS-PR) across these five runs.

A **lower standard deviation** indicates greater stability and reproducibility, which is an important consideration for reliable anomaly detection in practical applications.

4.3 Univariate Dataset Analysis

Table 1: Performance of Anomaly Detection Models on the Univariate Dataset

Algorithm	AUC-ROC	VUS-PR	Standard-F1	PA-F1
DLinear	0.6676 ± 0.0056	0.0516 ± 0.0030	0.0957 ± 0.0087	0.9893 ± 0.0043
KMeansAD_U	$\mathbf{0.9974 \pm 0.0001}$	$\mathbf{0.9712 \pm 0.0016}$	$\mathbf{0.8459 \pm 0.0043}$	$\mathbf{1.0000 \pm 0.0000}$
LSTMAD	0.5434 ± 0.0260	0.0290 ± 0.0027	0.0460 ± 0.0058	0.7696 ± 0.0752
OmniAnomaly	0.4837 ± 0.0018	0.0247 ± 0.0001	0.0484 ± 0.0008	0.4102 ± 0.0106
Sub_LOF	0.8525 ± 0.0000	0.7049 ± 0.0000	0.7407 ± 0.0000	0.9816 ± 0.0000
Sub_OCSVM	0.5245 ± 0.0000	0.0272 ± 0.0000	0.0500 ± 0.0000	0.6416 ± 0.0000
Sub_PCA	0.3215 ± 0.0000	0.0254 ± 0.0000	0.0558 ± 0.0000	0.3550 ± 0.0000
USAD	0.4768 ± 0.0430	0.0205 ± 0.0023	0.0416 ± 0.0050	0.3119 ± 0.1499

4.3.1 Model Performance Analysis

As summarized in Table 1, KMeansAD consistently rank top across all metrics

AUC-ROC: `KMeansAD_U` achieved the highest performance with a mean score of 0.9974 and an extremely low standard deviation of 0.0001, indicating excellent stability. `Sub_LOF` ranked second with a mean of 0.8525 and zero standard deviation, also showing high stability. `DLinear` came third (mean = 0.6676, std = 0.0056). In contrast, `Sub_PCA` performed the worst (mean = 0.3215).

VUS-PR: `KMeansAD_U` again performed best (mean = 0.9712, std = 0.0016). `Sub_LOF` also performed well (mean = 0.7049, std = 0.0000). Other algorithms achieved low scores, all below 0.1.

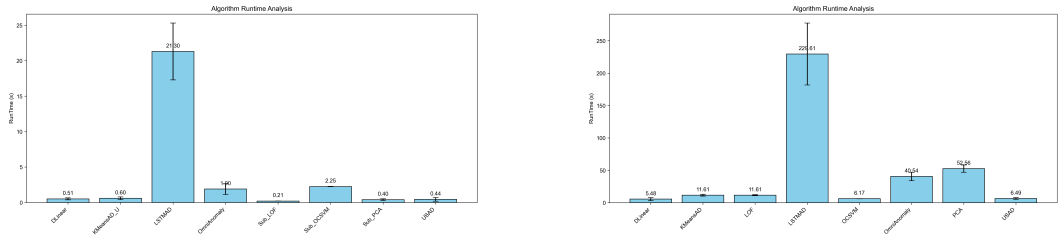
Standard-F1: `KMeansAD_U` significantly outperformed others with a mean of 0.8459. `Sub_LOF` followed (mean = 0.7407). Other methods had poor performance, with means below 0.1.

PA-F1: `KMeansAD_U` reached a perfect score of 1.0 with zero variance. `DLinear` also performed well (mean = 0.9893), followed by `Sub_LOF` (mean = 0.9816). `Sub_PCA` had the lowest score (mean = 0.3550).

4.3.2 Model Stability Analysis

`KMeansAD_U`, `Sub_LOF`, `Sub_OCSVM`, and `Sub_PCA` showed excellent stability across all metrics, particularly the latter three with zero standard deviation. In contrast, `USAD` had high variance, especially in PA-F1 (std = 0.1499) and AUC-ROC (std = 0.0430). `LSTMAD` also showed considerable fluctuation in AUC-ROC (std = 0.0260) and PA-F1 (std = 0.0752).

4.3.3 Runtime Analysis



(a) Univariate Dataset

(b) multivariate Dataset

Figure 1: Runtime comparison of two models.

As illustrated in Figure 1, the runtime characteristics vary significantly between univariate and multivariate settings.

Fastest Algorithms: Sub_LOF was the fastest, with an average runtime of 0.21 seconds and a very small standard deviation of 0.006 seconds. Sub_PCA (0.40s), USAD (0.44s), and DLinear (0.51s) also showed low runtime. KMeansAD_U had a moderate runtime of 0.60 seconds.

Slowest Algorithms: LSTMAD was the slowest, taking 21.30 seconds on average with a large standard deviation of 4.01 seconds. OmniAnomaly followed at 1.90 seconds, while Sub_OCSVM required 2.25 seconds.

Stability: Sub_LOF and Sub_OCSVM were the most stable in runtime (std = 0.006s and 0.025s, respectively). USAD and DLinear showed larger variability (std = 0.26s and 0.15s), while LSTMAD was both slow and unstable (std = 4.01s).

4.4 Multivariate Dataset Analysis

Table 2: Performance of Anomaly Detection Models on the Multivariate Dataset

Algorithm	AUC-ROC	VUS-PR	Standard-F1	PA-F1
DLinear	0.6249 ± 0.0014	0.2473 ± 0.0020	0.3472 ± 0.0029	0.7086 ± 0.0280
KMeansAD	0.7311 ± 0.1671	0.2884 ± 0.0761	0.4139 ± 0.0983	0.9438 ± 0.0019
LOF	0.5500 ± 0.0000	0.1708 ± 0.0000	0.2248 ± 0.0000	0.4390 ± 0.0000
LSTMAD	0.9957 ± 0.0013	0.9717 ± 0.0032	0.9538 ± 0.0036	0.9992 ± 0.0000
OCSVM	0.9378 ± 0.0000	0.6446 ± 0.0000	0.8021 ± 0.0000	0.9123 ± 0.0000
OmniAnomaly	$\mathbf{0.9983 \pm 0.0000}$	$\mathbf{0.9782 \pm 0.0000}$	0.9809 ± 0.0000	0.9960 ± 0.0000
PCA	$\mathbf{0.9983 \pm 0.0000}$	$\mathbf{0.9782 \pm 0.0000}$	$\mathbf{0.9810 \pm 0.0000}$	0.9959 ± 0.0000
USAD	0.9944 ± 0.0008	0.9751 ± 0.0007	0.9782 ± 0.0005	$\mathbf{0.9957 \pm 0.0000}$

4.4.1 Model Performance Analysis

As summarized in Table 2, PCA and OmniAnomaly consistently rank top across all metrics, while DLinear and LOF exhibit relatively weaker performance on multivariate data.

AUC-ROC: OmniAnomaly and PCA achieved the highest scores (mean = 0.9983, std = 0.0000). USAD and LSTMAD followed closely (mean = 0.9944 and 0.9957, respectively). DLinear and LOF underperformed (mean = 0.6249 and 0.5500).

VUS-PR: PCA and OmniAnomaly led the rankings (mean = 0.9782), followed by USAD (0.9751) and LSTMAD (0.9717). LOF had the lowest performance (mean = 0.1708).

Standard-F1: PCA, OmniAnomaly, and USAD all performed excellently (mean > 0.97). LSTMAD also showed strong performance (mean = 0.9538). DLinear, KMeansAD, and LOF had significantly lower performance (mean < 0.5).

PA-F1: All deep learning models (LSTMAD, OmniAnomaly, PCA, USAD) achieved high scores (mean > 0.99). KMeansAD also performed well (mean = 0.9438), while LOF scored the lowest (mean = 0.4390).

4.4.2 Model Stability Analysis

PCA, OmniAnomaly, OCSVM, and LOF were extremely stable, with near-zero or zero standard deviation across all metrics. In contrast, KMeansAD showed notable variance in AUC-ROC (std = 0.1671) and VUS-PR (std = 0.0761). DLinear also showed some fluctuation in PA-F1 (std = 0.0280).

4.4.3 Runtime Analysis

Fastest Algorithms: DLinear was relatively fast, averaging 5.48 seconds, though with a large standard deviation of 2.39 seconds. OCSVM and USAD also performed well, with runtimes of 6.17 and 6.49 seconds respectively. OCSVM was highly stable (std = 0.08s).

Slowest Algorithms: LSTMAD was by far the slowest (229.61s, std = 47.79s). PCA (52.56s) and OmniAnomaly (40.54s) also had high runtimes. KMeansAD and LOF were moderate (11.6s each).

Stability: OCSVM had the most consistent runtime (std = 0.08s), followed by LOF (std = 0.79s). LSTMAD was the most unstable (std = 47.79s). DLinear was fast but unstable (std = 2.39s).

4.5 Comparative Insights and Conclusion

Univariate vs. Multivariate Performance: Deep learning models performed significantly better on multivariate datasets than on univariate ones. Clustering-based methods such as KMeansAD_U performed best on univariate datasets. LOF-based methods underperformed in multivariate scenarios, while their variant Sub_LOF excelled on univariate data.

Algorithm Stability: Distance-based and subspace methods (Sub_LOF, Sub_OCSVM, Sub_PCA) were more stable on univariate datasets. On multivariate datasets, deep learning models (PCA, OmniAnomaly) demonstrated superior stability. KMeansAD had relatively poor stability in the multivariate setting.

Runtime Scalability: All models showed increased runtime on multivariate datasets. LSTMAD had the worst scalability, increasing from 21.30s (univariate) to 229.61s (multivariate). DLinear also saw a tenfold increase (0.51s to 5.48s). Subspace methods were more efficient in univariate settings.

Recommendations: For univariate time series anomaly detection, KMeansAD_U and Sub_LOF are recommended due to their balanced performance and efficiency. For multivariate time series, PCA and OmniAnomaly offer top-tier accuracy and stability. DLinear and USAD provide a good trade-off between performance and runtime.

Efficiency vs. Performance Trade-off: While deep learning models achieve excellent performance on multivariate datasets, they are computationally expensive. LSTMAD, despite its strong performance, may be unsuitable for real-time or resource-constrained applications. Simpler models like KMeansAD and OCSVM offer practical alternatives with faster execution and acceptable accuracy.

This analysis reinforces that there is no universal best anomaly detection algorithm. Model selection should be driven by the specific application scenario, data characteristics, and trade-offs between accuracy and computational efficiency.

5 Conclusion

In this work, we conducted a thorough empirical evaluation of unsupervised time series anomaly detection models using two representative datasets. Our experiments covered a broad spectrum of model types, including classical clustering-based methods, deep learning reconstruction models, and lightweight forecasting architectures. By incorporating standardized tuning procedures and robust evaluation metrics from the TSB-AD benchmark, we ensured consistent and fair comparison across models.

Our findings suggest that no single method dominates across all settings. On univariate data, clustering-based models such as KMeansAD_U and Sub_LOF deliver excellent performance with low runtime and high stability. In contrast, statistic model like PCA and deep learning-based methods such as OmniAnomaly excel in multivariate settings, offering top-tier accuracy but at a higher computational cost. DLinear, the proposed model, demonstrates a strong balance between efficiency and accuracy, particularly in runtime-constrained environments.

We emphasize that model selection should consider both the data characteristics and the deployment context. Future work may explore improving model scalability, integrating interpretability into deep anomaly detectors, and developing adaptive methods capable of handling concept drift in streaming data environments.

References

[1] Aggarwal , C. C. An introduction to outlier analysis. *Outlier analysis*, pages 1–34. 2017, .

[2] Aggarwal , C. C. (2017,) *An Introduction to Outlier Analysis*. *An Introduction to Outlier Analysis*: Springer.

[3] Chauhan , S. & Vig , L. (2015) Encdec-ad: Anomaly detection on multivariate time series using encoder-decoder neural networks. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* pages 1–7. IEEE.

[4] Dau , H. A., Keogh , E., Kamgar , K., Yeh , C.-C. M., Zhu , Y., Gharghabi , S., Ratanamahatana , C. A., Yanping , Hu , B., Begum , N., Bagnall , A., Mueen , A., Batista , G., & Hexagon-ML . 2018. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`.

[5] Ding , N., Ma , H., Gao , H., Ma , Y., & Tan , G. (2019) Real-time anomaly detection based on long short-term memory and gaussian mixture model. *Computers & Electrical Engineering* **79**:106458.

[6] Dodge , Y. (2008) *Time Series*. New York, NY: Springer New York.

[7] Fawcett , T. (2006) An introduction to roc analysis. *Pattern recognition letters* **27**(8):861–874.

[8] Hadjem , M., Naït-Abdesselam , F., & Khokhar , A. A. (2016) St-segment and t-wave anomalies prediction in an ecg data using rusboost. In *Healthcom*

[9] Hawkins , D. M. (1980,) *Identification of outliers*. London: Chapman and Hall.

[10] Hawkins , D. M. (1980,) *Identification of Outliers*, *11*. *11*: Springer.

[11] Jacob , V., Song , F., Stiegler , A., Rad , B., Diao , Y., & Tatbul , N. (2021) Exathlon: A benchmark for explainable anomaly detection over time series. *arXiv preprint arXiv:2010.05073*

[12] Kim , S., Choi , K., Choi , H.-S., Lee , B., & Yoon , S. (2022) Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence 36*, pp. 7194–7201.

[13] Li , D., Chen , D., Jin , B., Shi , L., Goh , J., & Ng , S.-K. (2019) Multivariate anomaly detection for time series data with generative adversarial networks. In *ICANN* pages 703–716. Springer.

[14] Liu , Q. & Paparrizos , J. (2024) The elephant in the room: Towards a reliable time-series anomaly detection benchmark. *Advances in Neural Information Processing Systems (NeurIPS)*

[15] Malhotra , P., Vig , L., Shroff , G., & Agarwal , P. (2015) Long short term memory networks for anomaly detection in time series. In *ESANN 89*, pp. 89–94.

[16] Masud , M. M., Chen , Q., Khan , L., Aggarwal , C., Han , J., & Thuraisingham , B. (2010) Addressing concept evolution in concept-drifting data streams pages 929–934.

[17] Munir , M., Siddiqui , S. A., Dengel , A., & Ahmed , S. (2018) Deepant: A deep learning approach for unsupervised anomaly detection in time series. In *IEEE Access 7*, pp. 1991–2005.

[18] Paparrizos , J., Boniol , P., Palpanas , T., Tsay , R. S., Elmore , A. J., & Franklin , M. J. (2022) Volume under the surface: A new accuracy evaluation measure for time-series anomaly detection. *Proceedings of the VLDB Endowment* **15**(11):2774–2787.

[19] Xu , H., Chen , W., Li , Z., Pei , D., Chen , J., Qiao , H., Feng , Y., & Wang , Z. (2019) Unsupervised anomaly detection on seasonal kpis with diverse seasonal patterns. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications* pages 1891–1899. IEEE.

[20] Xu , J., Wu , H., Wang , J., & Long , M. (2021) Anomaly transformer: Time series anomaly detection with association discrepancy. In *ICLR*

[21] Yue , Z., Wang , Y., Duan , J., Yang , T., Huang , C., Tong , Y., & Xu , B. (2022) Ts2vec: Towards universal representation of time series. In *AAAI 36*, pp. 8980–8987.

[22] Zeng , A., Zhang , Z., Zheng , Y., Liu , W., Xu , X., Zhang , Q., & al. (2023) Are transformers effective for time series forecasting? *Advances in Neural Information Processing Systems (NeurIPS)*

[23] Zhang , X., Zhao , Z., Tsiligkaridis , T., & Zitnik , M. () Self-supervised contrastive pre-training for time series via time-frequency consistency. In *NeurIPS* page 398.

[24] Zhang , Y., Wang , Y., Chen , J., Yu , H., & Qin , T. (2021) Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In *KDD* pages 3220–3230.