AutoTSAD: A Comprehensive Study of Automated Solutions for Time Series Anomaly Detection [E, A, & B]

Qinghua Liu
The Ohio State University
Columbus, OH, USA
liu.11085@osu.edu

Seunghak Lee
Meta
Menlo Park, CA, USA
seunghak@meta.com

John Paparrizos
The Ohio State University
Columbus, OH, USA
paparrizos.1@osu.edu

ABSTRACT

Time series anomaly detection is a fundamental data analytics task across scientific fields and industries. Despite recent attention, the effectiveness of the proposed anomaly detectors is restricted to specific domains. It is worth noting that a model that performs well on one dataset may not perform well on another. Therefore, how to select the optimal model for a particular dataset has emerged as a pressing issue. However, there is a noticeable gap in the existing literature when it comes to providing a comprehensive review of the ongoing efforts in this field. The evaluation of proposed methods across different datasets and under varying assumptions may create an illusion of progress. To date, there is no systematic evaluation conducted to assess the performance of these methods relative to each other. In this study, we (i) review the existing literature on automated anomaly detection and provide a taxonomy; (ii) introduce a comprehensive benchmark AutoTSAD which comprises 18 different methods and 60 variants; (iii) conduct a systematic evaluation on 1918 different time series across 18 datasets. Our study uncovers a significant gap, where over half of the proposed solutions to date do not statistically outperform a simple random choice. We also reveal previously overlooked yet highly accurate solutions. Moreover, we identify the challenges faced by existing approaches and outline potential research directions. To foster the development of new emerging solutions, we open-source our benchmark. Our aim for this study is to act as a catalyst, steering research efforts towards automated solutions in time series anomaly detection.

PVLDB Reference Format:
Qinghua Liu, Seunghak Lee, and John Paparrizos. AutoTSAD: A Comprehensive Study of Automated Solutions for Time Series Anomaly Detection [E, A, & B]. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/TheDatumOrg/AutoTSAD.

1 INTRODUCTION

Advances in cost-effective sensing, networking, storage, and processing technologies have facilitated the collection of vast amounts of sequential measurements over time. These ordered sequences of real-valued observations are commonly referred to as *time series*. Time series analysis has emerged as a field of significant interest, offering critical insights into a wide range of natural and human-driven phenomena. A wide array of time series mining tasks, including classification, forecasting, and anomaly detection, has been explored in the literature [27, 71]. Time series anomaly detection, which describes the process of analyzing an instance to identify abnormal patterns, has become critical across multiple scientific fields and industries. The presence of anomalies can indicate novel or unexpected events, such as imperfections in measurement systems and potential interactions with malicious entities. The applications span diverse areas including fraud detection in financial markets [11], network intrusion detection [55], as well as Key Performance Indicators (KPIs) monitoring in web applications [94].

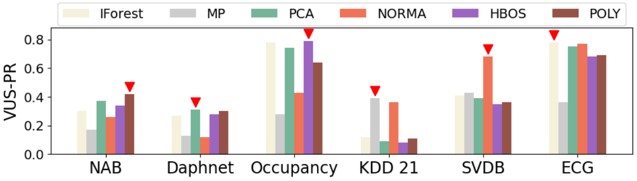


Figure 1: Detection accuracy (VUS-PR) for 6 anomaly detectors across different datasets within the TSB-UAD benchmark [73]. The red triangle indicates the model with the best detection accuracy: different winners in each dataset.

Recent years have witnessed a surge in the development of anomaly detection algorithms. Previous studies have evaluated the performance of these methods across different datasets [38, 73, 81]. These investigations have consistently highlighted the absence of a one-size-fits-all anomaly detector, as depicted in Figure 1. The scarcity of labeled data in anomaly detection has led researchers to prefer unsupervised approaches. These approaches are based on specific assumptions about data anomalies, resulting in performance variations when applied to heterogeneous time series datasets. As a result, achieving optimal performance requires in-depth knowledge of the myriad of methods. This necessity drives data analysts into an exhaustive and time-consuming process of trial-and-error procedure, making it cumbersome to select suitable anomaly detectors and fine-tune parameters to attain effectiveness.

The primary question that arises is: How can we **automatically** identify the most accurate anomaly detector in an **unlabeled** time series dataset? Here, the term ‘automated’ refers to eliminating the need for human expertise in the model selection process, and ‘unlabeled’ indicates the absence of any pre-existing labels that can be used as a reference during inference. The focus of this study is to investigate the effectiveness of automated solutions targeting

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

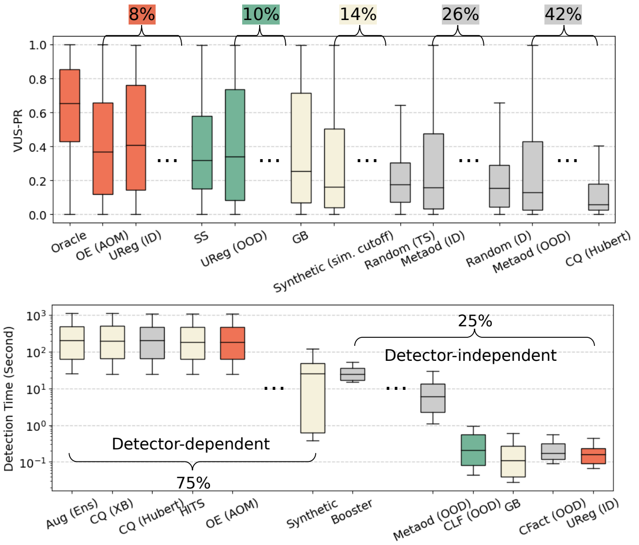


Figure 2: Overview of automated solutions for time series anomaly detection in terms of accuracy (top) and efficiency (bottom). Methods are arranged from left to right based on their performance, with the highest accuracy (measured by VUS-PR) on the left and the shortest detection time on the right. These methods are grouped into different clusters, with the ratio indicating the number of methods in each cluster.

the testing phase, without disregarding some methods that might benefit from leveraging historical knowledge.

Several challenges emerge regarding automated anomaly detection. First, the absence of labeled data hinders the accurate evaluation and comparison of different models, limiting effective model validation and selection. Second, the lack of a universal and widely accepted objective criterion (i.e., loss function) further complicates the comparison of different anomaly detection models. Additionally, the variety in datasets and anomaly types necessitates the need for adaptable evaluation metrics, making the model comparison more complex. Automated machine learning (AutoML) techniques [39] have emerged as promising approaches for model selection and generation without human intervention. However, most methods in the AutoML field require access to ground truth labels, making them unfeasible for our problem settings.

There have been some attempts made to address such challenges [35, 60, 86]. However, these studies exhibit certain limitations. Specifically, Ma *et al.* [60] provide an evaluation of unsupervised model selection for anomaly detection, yet their analysis primarily revolves around internal evaluation methodologies. Goswami *et al.* [35] target the time series scenario; however, their approach also confines itself to internal evaluation. Conversely, Sylligardos *et al.* [86] explore pretraining-based techniques, but their study is only focused on model selection via pretraining-based classifiers. The variability in datasets and different assumptions regarding application scenarios in these studies present a significant challenge when attempting to conduct a meta-analysis of their empirical performance. What is more, the efficacy of automated solutions in the context of time series datasets remains insufficiently validated.

To tackle the outlined problems and gain insights into the current state of research in this domain, we present the most extensive evaluation study to date, encompassing a broad array of automated solutions. By reviewing literature from the past decades on automated anomaly detection efforts, we establish a detailed taxonomy and develop an end-to-end benchmark. In Figure 2, we present an overview of the performance of automated solutions. The detection accuracy is categorized into five clusters based on five baselines. Out of 60 variants in the benchmark, only 8% of the methods exceed *Supervised Selection (SS)*, which represents the current practice involving labeling a subset of data and utilizing the best performing model for the rest of the dataset. When comparing the performance of pretraining-based model selector, there is a significant gap in performance from the in-distribution case to the out-of-distribution case, highlighting the need for domain generalization. Additionally, more than 60% of the methods do not exhibit advantages over a simple random choice. For detection time, these solutions are divided into two clusters based on whether they require anomaly scores that are generated from the complete set of candidate models. We have identified completely unsupervised solutions that show high accuracy but require a considerable amount of detection time, and pretraining-based solutions demonstrate superiority in accuracy and efficiency but require pretraining on historical labeled datasets. Further elaboration will be provided in subsequent sections.

In our study, we identify three principal design dimensions for addressing automated anomaly detection problems. These design dimensions also serve as a guide for setting up the evaluation pipeline. The first dimension represents the initial stage in automated solutions, which involves constructing *Candidate Model Set.* In addition to focusing solely on selecting optimal model hyperparameters, we revisit and include the data handling pipeline of time series anomaly detection as an additional selection dimension. This approach, in line with the concept introduced in a recent study [44], enhances the model selection process by incorporating a range of representative design choices. In the second dimension, *Automated Detection Pipeline*, we integrate principal design choices from diverse methodological categories and various use case assumptions into a single testbed, allowing for a fair and direct comparison of their performance. The third dimension, *Performance Evaluation*, concentrates on evaluating the effectiveness of the methods in terms of accuracy and detection time. To measure accuracy, we use the latest anomaly detection evaluation metrics specifically tailored for time series data [72] and conduct evaluation across 18 different datasets from diverse time series fields to ensure a robust and convincing evaluation. For the detection time analysis, we decompose the pipeline into different components to provide a detailed overview.

We start with a discussion of related work for time series anomaly detection (Section 2.1). Then, we present our contributions:

- We review the existing literature on automatic anomaly detection and formulate a taxonomy (Section 2.2).
- We introduce the key methodologies identified in taxnomomy that are suitable for our problem settings (Section 3).
- We develop an end-to-end benchmark AutoTSAD that encapsulates the primary categories of automated solutions (Section 4).
- We carry out a systematic evaluation of 18 solutions along with 60 variants on a large scale of time series datasets (Section 5).
- We offer recommendations on the existing works and outline potential future research directions (Section 6).

Finally, we conclude with the implications of our work (Section 7).

2 RELATED WORK

In this section, we first review relevant research on time series anomaly detection (Section 2.1) and then present the our proposed taxonomy for automated anomaly detection (Section 2.2).

2.1 Time Series Anomaly Detection

We begin with the definitions of anomaly in the context of time series and then proceed to introduce different categorizations of anomaly detectors. Finally, we introduce the detection pipeline.

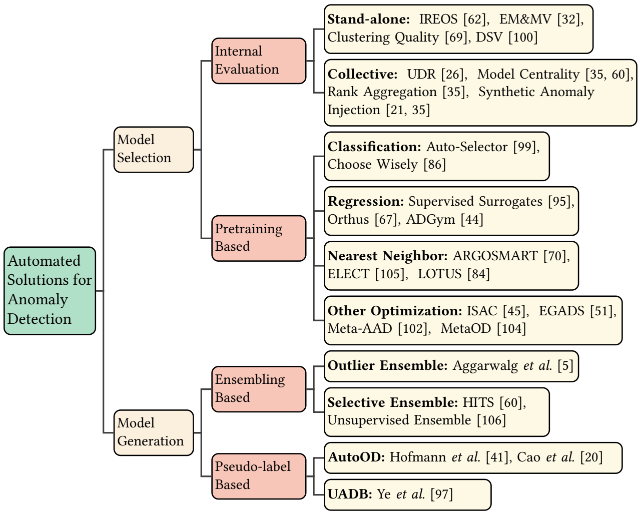
Definition: Anomalies in time series can occur in the form of a single value or collectively in the form of sub-sequences. Formally, they can be categorized into three types: point, contextual, and collective anomalies. The first two categories, namely, point and contextual anomalies, are referred to as *point-based* anomalies. Collective anomalies are known as *sequence-based* anomalies. Point anomalies are individual data points that significantly deviate from the majority of the data. Contextual anomalies, in contrast to point anomalies, are data points that fall within the expected distribution range but diverge from the expected pattern in a given context (e.g., within a time window). Collective anomalies refer to sequences of points that deviate from a typical, previously observed pattern.

Category of Method: The approaches to this task can be categorized based on the level of prior knowledge available: (i) unsupervised, which does not require any labeled data; (ii) semi-supervised, requiring labels only for normal instances; and (iii) supervised, which requires a labeled dataset containing both normal and anomalous instances. In practical applications, due to the limited availability of labeled anomalies, unsupervised or semi-supervised anomaly detection methods are more feasible. Based on the nature of the processing, the methods can be divided into three categories: (i) distance-based methods, which analyze subsequences to detect anomalies in time series, primarily by calculating distances to a given model [12, 16]; (ii) density-based methods, identify anomalies by focusing on isolated behaviors within the overall data distribution, rather than measuring nearest-neighbor distances [4, 57]; and (iii) prediction-based methods, which propose to train a model on anomaly-free time series and then reconstruct the data or forecast future points [65, 78]. In this way, the anomalies are identified by significant deviations between predictions and the actual data.

Detection Pipeline: The general pipeline for time series anomaly detection comprises three stages: preprocessing of time series data, application of an anomaly detector, and post-processing of the anomaly scores. In the first stage, the time series is transformed into a matrix format, with each row representing sliding window slices of the original series. Subsequent to this transformation, various anomaly detectors can be applied to the windowed data and generate anomaly scores that indicate the level of abnormality for each data point. A higher value of anomaly score suggests a higher probability of abnormality. During the final stage of post-processing, a threshold is commonly employed to determine anomalies in the time series. This is done by comparing the anomaly score of each data point against the predefined threshold value.

2.2 AutoML for Anomaly Detection

AutoML presents a promising approach to constructing machine learning systems without human intervention [42]. The challenge is formally defined as the Combined Algorithm Selection and Hyperparameter (CASH) problem. Several successful studies have been conducted to tackle this issue [7, 29, 88]. The process involves a range of tasks, such as feature selection, feature extraction, model selection, and hyperparameter tuning. The evaluation of model performance is carried out using predetermined quality metrics, such as accuracy for classification tasks and the silhouette measure for clustering tasks. However, it has been noted that the primary focus within the AutoML community is largely centered around supervised learning. In contrast, the applicability to unsupervised learning scenarios, particularly in anomaly detection, is relatively limited due to challenges such as the absence of labeled data and the lack of well-defined quality metrics for unsupervised approaches [9]. Some supervised search methods proposed for anomaly detection use metrics like F-score or ROC as quality measures [49, 52, 53, 103], but their applicability remains limited since comparing different models still necessitates the availability of ground truth labels.



Automated Solutions for Anomaly Detection
Model Selection
Internal Evaluation
Stand-alone: IREOS [62], EM&MV [32], Clustering Quality [69], DSV [100]
Collective: UDR [26], Model Centrality [35, 60], Rank Aggregation [35], Synthetic Anomaly Injection [21, 35]
Pretraining Based
Classification: Auto-Selector [99], Choose Wisely [86]
Regression: Supervised Surrogates [95], Orthus [67], ADGym [44]
Nearest Neighbor: ARGOSMART [70], ELECT [105], LOTUS [84]
Other Optimization: ISAC [45], EGADS [51], Meta-AAD [102], MetaOD [104]
Model Generation
Ensembling Based
Outlier Ensemble: Aggarwalg et al. [5]
Selective Ensemble: HITS [60], Unsupervised Ensemble [106]
Pseudo-label Based
AutoOD: Hofmann et al. [41], Cao et al. [20]
UADB: Ye et al. [97]

Figure 3: Taxonomy of automated anomaly detection solutions, of which operate without supervision during test time.

In recent decades, multiple solutions have been proposed to tackle the aforementioned challenges. We present a taxonomy of existing automated solutions in anomaly detection as depicted in Figure 3. From a process-centric perspective, the works in this field can be classified into two main categories, namely model selection and model generation. **Model selection** refers to identifying the optimal model and its corresponding hyperparameters from a predefined set. Subsequently, the selected model is utilized for anomaly detection. On the other hand, **Model generation** entails the construction of a completely new model based on the predefined set. This newly generated model then can operate independently as an anomaly detector. Within the model selection category, the existing literature can be further categorized into two groups: Internal Evaluation and Pretraining-based methods. The former evaluate the effectiveness of a model by using surrogate metrics for anomaly detection, independent of external data such as ground truth labels for anomalies [21, 26, 32, 35, 60, 62, 69, 100]. On the other hand, Pretraining-based methods leverage the knowledge of performance of various anomaly detectors on historical

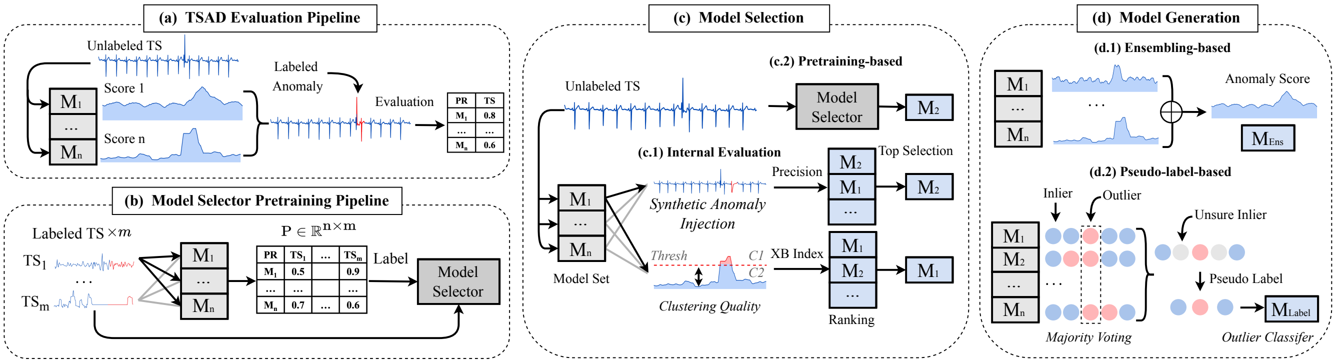


Figure 4: Overview of AutoTSAD benchmark. We use M_1, M_2, and M_n to represent the candidate models. (a) depicts the standard evaluation pipeline for anomaly detectors. (b) depicts the pretraining pipeline for pretraining-based model selectors. (c) outlines the process for Model Selection which includes two main categories: (c.1) Internal Evaluation (Section 3.1.1) and (c.2) Pretraining-based Method (Section 3.1.2). The output is the chosen anomaly detector that can then be applied to the time series data. (d) shows the approach for Model Generation which includes: (d.1) Ensembling-based (Section 3.2.1) and (d.2) Pseudo-label-based Methods (Section 3.2.2). The result can be considered as an anomaly detector on its own.

labeled datasets to enable the automatic model selection for new datasets [44, 45, 67, 70, 86, 95, 104]. In the category of model generation, the existing literature can be categorized into Ensembling-based methods and Pseudo-label-based methods. Ensembling-based methods involve constructing models that combine ensembles from the predefined set [5, 60, 106]. Pseudo-label-based methods generate pseudo-labels to transform the unsupervised anomaly detection problem into a supervised framework [20, 97].

From the taxonomy presented in Figure 3, the objective of our work is to evaluate a wide range of solutions suitable for our problem settings across all major categories, which is guided by two main assumptions. The first assumption is that certain methods are not compatible with our benchmark due to various limitations. For instance, IREOS [62] requires a substantial amount of computational resources as it necessitates training a nonlinear classifier for each sample (in our case, a sample refers to a time series window), and itself introduces a model selection problem that the result may vary based on the chosen classifier and its hyperparameters. DSV [100] is designed for auto-encoder-based models, imposing restrictions on the diversity of candidate models. UDR is based on the assumption that a good model is expected to yield consistent results across various hyperparameter settings, a condition not always met in time series anomaly detection due to the variability in data and anomaly types (see examples in Section 4.2). EGADS [51], developed by Yahoo Lab for commercial applications, requires user feedback to determine the types of anomalies of interest. However, such user feedback is not available in our case, making it challenging to compare EGADS against other methods. The second assumption relates to methodologies in pretraining-based model selection, where we extract and re-implement the core concepts from each study, utilizing the same learning algorithms for fair comparisons. Building upon these two assumptions, in AutoTSAD benchmark, we have incorporated 18 different solutions referenced from 20 papers and integrated them into a unified framework. These methodologies will be elaborately discussed in the following Section 3.

3 AUTOTSAD BENCHMARK

In Figure 4, we present an overview of AutoTSAD benchmark. Given an unlabeled time series, Figure 4(a) illustrates the procedure for evaluating the performance of various anomaly detectors. Figure 4(c) demonstrates the model selection process, which will be elaborated on in Section 3.1, and Figure 4(d) represents the model generation process, which will be discussed further in Section 3.2.

3.1 Model Selection

The challenge of selecting an optimal anomaly detector from a predefined model set without supervision is defined as the Unsupervised Outlier Model Selection (UOMS) problem [35, 60, 104]. Methods in this category involve the development of surrogate metrics (Section 3.1.1) or the use of historical knowledge (Section 3.1.2).

3.1.1 Internal Evaluation. Internal evaluation methods evaluate the effectiveness of a model without any reliance on external information (i.e., ground truth labels for anomalies). As outlined by Ma et al. [60], and expanded upon in our work, these methods can be categorized into two groups: (i) *stand-alone*, which relies solely on each anomaly detector and its corresponding output anomaly score, and (ii) *collective*, which utilizes the interactions among models within the predefined candidate set. EM&MV and Clustering Quality are examples of stand-alone methods, whereas the remaining methods in the benchmark belong to the collective category. We then proceed to introduce these methods in the following.

EM&MV: To compare the performance of anomaly detectors without the need for labeled data, Goix [32] proposed two numerical performance criteria. These criteria are based on Mass-Volume [22] and Excess-Mass [33] curves. It eliminates the reliance of performance evaluation on Receiver Operating Characteristic (ROC) or Precision-Recall (PR) curves, which typically require labeled data.

Clustering Quality (CQ): Nguyen *et al.* [69] employ internal validation measures designed for clustering algorithms in anomaly detection. For this, anomaly scores can be partitioned into two clusters by setting thresholds (i.e., the abnormal points cluster and the normal points cluster). Subsequently, clustering metrics can be applied to assess their performance, determining the optimal model based on the assumption that an anomaly detector is considered 'good' when the two sets of scores are more distinctly separated and/or the scores within each set are more tightly clustered.

Model Centrality (MC): The concept of Model Centrality was initially introduced in self-supervised model selection for disentangling GANs [26, 56]. It is based on the hypothesis that well-disentangled models should approximate the optimal model and, consequently, exhibit proximity to one another. Subsequently, this approach has been adapted to the field of anomaly detection [35, 60], based on the assumption that there is one single ground truth, thus detectors close to this are likely close to each other. In this framework, the distance between two models is quantified using Kendall's τ distance, applied to the anomaly scores generated by models. The centrality of a model is thus defined as the average distance to its K nearest neighbors, where K is a predefined parameter. This metric is designed to favor models that are closely aligned with their nearest neighbors. However, a limitation of this metric arises from the potential clustering of poor detectors.

Synthetic Anomaly Injection (Synthetic): This method is based on the assumption that an effective anomaly detector should exhibit superior performance on data with artificially introduced anomalies [35]. The process involves the generation of synthetic datasets with anomalies, followed by an evaluation of models on these datasets. The model that exhibits the highest performance is then considered the optimal choice. Chatterjee *et al.* [21] propose a preliminary simulation protocol before the injection of anomalies. This protocol assumes that anomalies in actual time series typically appear in the trend component or as outliers in the residual component. In particular, they decompose the original time series with STL decomposition [23] and then construct the synthetic time series by adding the seasonality component and random noise with the same mean and standard deviation as the residual of STL. The injection of anomalies is based on the synthetic anomalies instead of the original time series as in Goswam *et al.* [35]. However, while simulated data provides valuable insights, it deviates from real-world scenarios, potentially leading to erroneous decisions.

Rank Aggregation (RA): The surrogate metrics mentioned above serve as a proxy of anomaly detection performance to some extent, yet they are not without imperfections. Goswami *et al.* [35] introduce a methodology for rank aggregation that accommodates the imperfect rankings indicated by these surrogate metrics. In their work, they explore the application of Kemeny rank aggregation [48], specifically employing an efficient approximation solution via the Borda method [14]. Additionally, they propose several robust variants of the Borda method, which focus on considering only the top k models and aggregating more reliable rankings.

3.1.2 Pretraining-based Method. Methods in this category require historical datasets annotated with anomalies, utilizing insights from these datasets to select the most appropriate model for new data. As depicted in Figure 4(b), a historical dataset with labeled anomalies $\mathrm{D_{train}} = \{\mathrm{D_1}, ..., \mathrm{D_m}\}$ is provided, where m represents the number of time series. Subsequently, a performance matrix $\mathrm{P} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$ is generated, with n indicating the count of models in the candidate set. The matrix P is formulated by iteratively applying each anomaly detector from the candidate set to the labeled time series and performing evaluations. In this case, $\mathrm{P_{i,j}}$ corresponds to the j-th anomaly detector's performance on the i-th historical dataset. The time series datasets serve as the training input, while the performance matrix functions as the target label. Given a new data $\mathrm{X_{New}} \in \mathbb{R}^{1\times\mathrm{T}}$, where T is the length of time series, the model selector identifies the best model among n candidate models. These methods can be further categorized based on the optimization function employed on the performance matrix as elaborated as follows.

Classification (CLF): The Classification-based method [86, 99] converts the model selection process into a classification task by training a classifier on $\mathrm{D_{train}}$ with m time series, each labeled with the best anomaly detector from n candidate models. For a new test time series $\mathrm{X_{New}}$, the model selector classifies it into one of n categories, effectively selecting the optimal anomaly detector.

Regression (RG): Methods in this group [44, 67, 95] frame model selection as a regression problem, using features of labeled data to predict the performance of each anomaly detector. The model selector (i.e., regressor) is optimized by mean squared error. In addition, Orthus [67] incorporates Singular Value Decomposition before regressor to control the regression complexity. For input $\mathrm{X_{New}}$, the model selector predicts the expected performance of each anomaly detector, choosing the one with the highest predicted performance.

Nearest Neighbor (kNN): These methods [70, 84, 105] find the closest train data $\mathrm{D_i}$ to the given $\mathrm{X_{New}}$ based on feature similarity and select the model with the best performance on $\mathrm{D_i}$. In particular, ELECT [105] calculates distances in performance space to find the closest matches to $\mathrm{X_{New}}$. And LOTUS [84] employs the Low-Rank GW approximation distance [80] for nearest neighbor searches.

Other Optimization: ISAC [45] clusters performance matrix P based on features. Given $\mathrm{X_{New}}$, it identifies its closest cluster and selects the best model within this cluster (i.e., the largest average performance on the cluster's datasets). MetaOD [104] is based on collaborative filtering: n models are evaluated over m different datasets, and a matrix factorization process approximates the performance of all models based on a projected matrix of meta-features extracted from the datasets. Given $\mathrm{X_{New}}$, its meta-features are extracted and then multiplied by the matrix factorization component, yielding a performance prediction for every model.

3.2 Model Generation

In contrast to model selection, which involves predicting the optimal model from the model set, model generation concentrates on creating an entirely new model tailored to a specific dataset based on the predefined model set. In our review of the literature in this category, we categorize the methodologies into two groups: ensembling-based methods and pseudo-label-based methods.

3.2.1 Ensembling-based Method. Ensemble learning integrates the informative knowledge from weak predictive results obtained from various learning algorithms (in this context, different anomaly detectors) to enhance knowledge discovery and predictive performance through adaptive voting schemes [25]. These methods can be broadly categorized based on their approach to ensemble set selection: outlier ensembles, which do not involve selection, and selective ensembles, which implement a level of selection.

Outlier Ensemble (OE): Aggarwal *et al.* [5] establish an analogy between outlier ensembles and the bias-variance theory in classification problems. They introduce three ensemble combination techniques: (i) *Average*, which computes the mean of scores from all anomaly detectors; (ii) *Maximization*, using the maximum score from all detectors for each data point; (iii) *Average of Maximum*

(AOM), averaging the maximum scores from a randomly chosen subset of detectors. The *Average* method is favored over *Maximization* as it tends to reduce variance, similar to the effect observed in classification problems, while *Maximization* may overestimate the absolute scores by picking out the larger errors. However, *Maximization* is advantageous for reducing bias, particularly in challenging datasets where outliers are not easily discernible and may receive inlier-like scores from many ensemble components. In these scenarios, outlier scores are often undervalued in comparison to inlier data points across most components. Utilizing a *Maximization* ensemble is an effective strategy for magnifying outlier-like behavior in specific components. Furthermore, *Average of Maximum* ensemble is built upon the above two methods and combines the merits of bias and variance reduction. These methods provide a strong baseline for time series anomaly detection scenarios.

Unsupervised Ensemble (UE): Zimek *et al.* [106] propose to obtain an ensembling anomaly score by aggregating outputs from a chosen subset of models. The process starts with the identification of a pseudo ground truth by averaging the anomaly scores in the candidate pool. Subsequently, the distance between it and each anomaly score in the candidate pool is calculated, and the closest anomaly score is chosen as the next pseudo ground truth. This process continues iteratively until a convergence criterion is met, at which point, the pseudo ground truth extracted from each iteration is averaged to serve as the final anomaly score.

HITS: Ma *et al.* [60] adapt the HITS algorithm [47], which was initially proposed for computing centrality in a network setting, for UMOS. In contrast to Model Centrality, which is computed in a single iteration, this approach proposes a recursive computation of centrality. The hubness centralities of candidate models can be used for evaluation and a model is considered more central or reliable if it directs (with a high anomaly score) to samples with high authority.

3.2.2 *Pseudo-label Based Method.*

Unsupervised anomaly detection does not require labeled data, but the accuracy of unsupervised techniques is often low due to the lack of supervision with domain knowledge [18]. On the contrary, supervised classification tends to achieve better accuracy, as long as a sufficient number of high-quality labels are available [4]. Instead of first carefully selecting an appropriate anomaly detection method and then tuning its parameters, a different approach involves generating pseudo labels to transform the unsupervised problem into a supervised one.

Label Set Augmentation (Aug): The fundamental insight behind this method [20, 41] is that selecting one model from many alternate unsupervised anomaly detectors may not always work well. Instead, it targets combining the best of them. This method begins by automatically identifying a small but reliable set of labels and iteratively augmenting this set through three steps: (i) *Initial Reliable Object Discovery*, where an initial set of outliers/inliers is determined through majority voting; (ii) *Learning-based Pruning of Poor Detector*, which uses these initial labels as pseudo-ground truth to prune less effective detectors via logistic regression, thereby refining the set of reliable labels. (iii) *Reliable Object Set Update*, which applies multi-view analysis [58] to refine the set of reliable objects based on comparisons between logistic regression outcomes and another classifier until a set of reliable objects does not change.

Table 1: Summary characteristics of the 18 datasets in the benchmark. 'Total' is the number of time series available in the dataset. 'Eva Count' and 'Anomaly Ratio' refer to statistics of the dataset in the evaluation set (see details in Section 4.1).

Dataset	Total	Eva Count	Anomaly Ratio	Description
Dodgers [43]	1	1	11.7%	unusual traffic after a Dodgers game
ECG [63]	52	26	7.5%	standard electrocardiogram dataset
IOPS [1]	58	29	1.9%	performance indicators of a machine
KDD21 [46]	247	124	0.5 %	UCR Anomaly Archive
MGAB [87]	10	5	0.2%	Mackey-Glass time series
NAB [6]	50	25	9.9%	web-related data
SensorScope [96]	23	12	22.1 %	environmental data
YAHOO [50]	319	160	0.7%	Yahoo production systems data
NASA-MSL [10]	24	12	7.9%	Curiosity Rover telemetry data
NASA-SMAP [10]	54	27	11.6%	Soil Moisture Active Passive satellite data
Daphnet [8]	45	23	9.2%	acceleration sensors
GHL [30]	126	63	0.2%	gasoil heating loop telemetry
Genesis [89]	6	3	0.3%	portable pick-and-place demonstrator
MITDB [64]	32	16	13.1%	ambulatory ECG recordings
OPP [76]	464	232	4.1%	motion sensors
Occupancy [19]	10	5	30.3%	room occupancy data
SMD [85]	281	141	3.8%	server machine telemetry
SVDB [36]	115	58	11.7%	ECG recordings

Label Set Cleaning (Clean): This method [20] starts with a large set of noisy labels and progressively cleans them to produce a more reliable set. The process involves three steps: (i) *Initial Training Data Generation*, marking all possible outliers determined by anomaly detectors as anomalies; (ii) *Modeling*, based on the assumption that model accuracy is higher for correctly labeled data early in the training phase [83], with ongoing loss tracking for each training instance; and (iii) *Training Data Update*, where data points associated with large early losses are excluded from the label set.

Unsupervised Booster (Booster): The objective of this method [97] is to develop a versatile booster model that improves the detection accuracy of any anomaly detectors by employing knowledge distillation. The primary focus is to move beyond static assumptions and empower the models with the ability to adapt to different datasets. Specifically, the method starts by distilling the knowledge of source anomaly detectors to a booster model and then exploiting the variance between them to perform automatic correction.

4 EXPERIMENTAL SETTINGS

In this section, we review the experimental settings of AutoTSAD. We begin by providing the setup of the benchmark (Section 4.1), followed by an introduction to the details of the candidate model set (Section 4.2) and automated solutions and baselines (Section 4.3). Lastly, we discuss the evaluation metrics employed (Section 4.4).

4.1 Experimental Setup

We now introduce the setup in terms of technical platform and implementation, along with the datasets we use as follows.

Platform: We conduct our experiments on a server with the following configuration: AMD EPYC 7713 64-Core. The server has two Nvidia A100 GPUs and runs Ubuntu 22.04.3 LTS (64-bit).

Implementation: We implemented the library and scripts that accompany AutoTSAD in Python 3.8 with the main following dependencies: Pytorch 1.11 [74], TensorFlow 2.6.0 [2], scikit-learn 0.22.1 [75]. For the anomaly detection methods, we used the implementation provided in the TSB-UAD benchmark [73]. For reproducibility purposes, we open-source the AutoTSAD benchmark[1].

Datasets: We use the 18 public datasets available in TSB-UAD

[1] https://github.com/TheDatumOrg/AutoTSAD

Table 2: Overview of the Candidate Model Set. A value of 1 in 'Win' indicates using the max periodicity of the time series as the sliding window length, and 2 denotes the second-max periodicity. A value of 0 implies that we do not apply the sliding window strategy. 'Model Hyperparameter' outlines the different hyperparameter settings (see TSB [73] for detailed definitions). We use a (Win, HP) tuple to specify hyperparameter configurations for each candidate model.

Method	Win	Model Hyperparameter	Candidate Model (Win, HP)
IForest	[0,1,2,3]	n_estimators=[20, 50, 75, 100, 150, 200]	(3, 200), (1,100), (0,200)
LOF	[1,2,3]	n_neighbors=[10, 30, 60]	(3,60), (1,30)
MP	[1,2,3]	cross_correlation=[False,True]	(2,False), (1,True)
PCA	[1,2,3]	n_components=[0.25, 0.5, 0.75, None]	(3,None), (1,0.5)
NORMA	[1,2,3]	clustering=[hierarchical, kshape]	(1,hierarchical), (3,shape)
HBOS	[1,2,3]	n_bins=[5, 10, 20, 30, 40, 50]	(3,20), (1,40)
POLY	[1,2,3]	power=[1, 2, 3, 4, 5, 6]	(3,5), (2,1)
OCSVM	[1,2,3]	kernel_set=[linear, poly, rbf, sigmoid]	(1,rbf), (3,poly)

AE	[1,2,3]	hidden_neuron=[[64, 32, 32, 64], [32, 16, 32]],
		norm=[bn, dropout]
			(1,[32, 16, 32],bn),
			(2, [64, 32, 32, 64],dropout)

CNN	[1,2,3]	num_channel=[[32, 32, 40], [8, 16, 32, 64]]
		activation=[relu, sigmoid, tanh]
			(2,[32, 32, 40],relu),
			(3,[8, 16, 32, 64],sigmoid)

LSTM	[1,2,3]	hidden_dim=[32, 64],
		activation=[relu, sigmoid]
			(1,64,relu),
			(3,64,sigmoid)

benchmark [73], which consist of 1918 time series spanning diverse domains. For evaluation purposes, each dataset is split into two segments: the *label set* with ground truth anomaly labels and *evaluation set* without labels. In cases where a dataset contains only one instance, we partition the entire time series into two parts, ensuring that each part contains anomalies and include the first half in the label set and the second half in the evaluation set. Consequently, the labeled set comprises a total of 956 time series, while the evaluation set encompasses 962 time series in total. The description of the data in the evaluation set is available in Table 1. All evaluations of automated solutions are conducted exclusively on the evaluation set, with the labeled set serving as the resource for supervised selection and pretraining data for pretraining-based model selectors.

4.2 Candidate Model Set

We proceed to introduce the candidate models and their corresponding hyperparameter settings in the candidate model set.

Predifined Model Set: We select 11 different anomaly detection methods, covering the main categories of time series anomaly detection algorithms introduced in Section 2.1. IForest [57] constructs the binary tree, wherein the path length from the root to a node serves as an indicator of anomaly likelihood; shorter paths suggest higher anomaly probability. LOF [17] calculates the anomaly score by comparing local density with that of its neighbors. HBOS [34] constructs a histogram for the data and uses the inverse of the height of the bin as the anomaly score of the data point. MP [98] identifies anomalies by pinpointing the subsequence exhibiting the most substantial nearest neighbor distance. OCSVM [82] fits the dataset to find the normal data's boundary. NORMA [13] identifies the normal pattern based on clustering and calculates each point's effective distance to the normal pattern. PCA [3] projects data to a lower-dimensional hyperplane, with significant deviation from this plane indicating potential outliers. AE [79] projects data to the lower-dimensional latent space and then reconstructs it, where anomalies are typically characterized by evident reconstruction deviations. LSTM-AD [61], POLY [54], and CNN [66] model the relationship between current and preceding time series data, detecting anomalies through discrepancies between predicted and actual values. Among the methods, seven fall under the unsupervised category, indicating that they can identify anomalies without the need for any prior knowledge. These methods consist of IForest, LOF, MP, NORMA, PCA, HBOS, and POLY. The remaining four methods are considered semi-supervised as they require a certain level of prior information regarding normal behavioral patterns. Following common practice in TSB [73], we train these models on the initial regions of the time series. We expect that anomalies with a low density (< 5%) in the training data would not affect the result.

Hyperparameter Selection in Model Set: As depicted in Table 2, we analyze two aspects of hyperparameters: the sliding window length in time series preprocessing and the model hyperparameter. The sliding window length serves as a critical yet previously overlooked hyperparameter, utilized for converting time series data into a windowed format. We determine the window length based on the time series periodicity which can be estimated using the autocorrelation function. Given that several time series exhibit multiple periodicities, we consider the max, second, and third max periodicities as three potential options. The detailed model hyperparameters for each anomaly detector are availe in Table 2. In this way, we obtain a total of 159 different models. However, it is not realistic to carry out experiments on such a large number of candidates, and it is essential to acknowledge that certain hyperparameters are not valid in the context of time series and should be disregarded. Building upon this notion, we propose a method to prune this set and obtain a more manageable and reliable candidate set. In order to identify high-quality candidate models, we consider two main factors. The first factor focuses on ensuring that the model, under a specific hyperparameter setting, exhibits overall robustness and accuracy. The second factor acknowledges that while a model may not consistently perform optimally, it should demonstrate the ability to outperform the model selected in the previous step, both in terms of the number of wins and the extent of superiority. We assess the validity of hyperparameter combinations on a randomly selected subset of datasets from TSB. By considering these two factors (i.e., selecting the best model and the most complementary model), we can refine our initial set of 159 models down to a set of 23 models. Each anomaly detector in this refined set is associated with 2-3 hyperparameter combinations. This approach reduces computational complexity and enhances the meaningfulness of model selection.

4.3 Benchmark Details

In this section, we first introduce our baselines for comparison and then details of automated solutions in the benchmark.

Baseline: We employ four types of baselines. The **Oracle** represents the theoretical upper bound for model selection, where the most accurate anomaly detector for a time series is selected based on ground truth labels. **Global Best (GB)** selects the model that exhibits the highest overall performance (highest average ranking) across the entire evaluation set. **Supervised Selection (SS)** identifies the best anomaly detector on the label set of each dataset and then uses it for the evaluation set. Compared with GB, which selects a single model globally, SS identifies the best model for each dataset, resulting in a total of 18 selected models for 18 datasets. Random Choice simulates the model selection process absent of any prior knowledge or expertise. We consider two variants of random

Table 3: Overview of AutoTSAD benchmark. 'Variants' indicate different variations for each method. 'TS' indicates whether the method is proposed for the time series scenario. 'D' indicates whether it requires anomaly scores generated from the complete candidate model set. And 'S' indicates the requirement of supervision from pretraining data.

Method	Reference	Variants	TS	D	S
EM&MV	[32]	[Excess-Mass, Mass-Volume]×2	×	✓	×
CQ	[69]	[XB, Silhouette, R2, ...]×10	×	✓	×
MC	[35, 60]	[1N, 3N, 5N]×3	✓	✓	×
Synthetic	[21, 35]	[sim. cutoff, orig. cutoff, ...]×12	✓	✓	×
RA	[35]	[Borda, Trimmed Borda]×6	✓	✓	×
CLF	[86, 99]	[ID, OOD]×2	✓	×	✓
RG	[44, 95]	[ID, OOD]×2	×	×	✓
UReg	[67]	[ID, OOD]×2	✓	×	✓
CFact	[67]	[ID, OOD]×2	✓	×	✓
kNN	[70, 84, 105]	[ID, OOD]×2	×	×	✓
MetaOD	[104]	[ID, OOD]×2	×	×	✓
ISAC	[45]	[ID, OOD]×2	×	×	✓
OE	[5]	[Avg, Max, Avg of Max]×3	×	✓	×
UE	[106]	1	×	✓	×
HITS	[60]	1	×	✓	×
Aug	[20, 41]	[Majority Voting, Orig, Ens]×3	×	✓	×
Clean	[20]	[Majority, Individual, Ratio, Avg]×4	×	✓	×
Booster	[97]	1	×	×	×
Count: 18	**20**	**60**			

choice: (i) **Random (D)**, where a model is randomly selected from the candidate set for each dataset and utilized for anomaly detection on that dataset; (ii) **Random (TS)**, where a model is randomly chosen for each time series and then applied on that.

Internal Evaluation: As depicted in 3, our benchmark comprises a total of 18 methods and 60 variants. For EM&MV, we adopt two unsupervised criteria from Goix [32], which are based on the Excess-Mass and Mass-Volume curves. For CQ, our study encompasses 10 different measures of clustering quality, including the Xie-Beni index [93] and the Silhouette index [77], etc. For MC, we set the number of nearest neighbors to 1, 3, and 5 when calculating the average distance between anomaly scores. Regarding Synthetic, we categorize the methods into two primary groups: one utilizing the original time series and another applying a preliminary simulation protocol as described by Chatterjee *et al.* [21]. For anomaly injection techniques, we draw upon strategies from Goswami *et al.* [35], such as cutoff, speedup, etc. We denote methods using the original time series as 'orig.' and those using synthetic time series as 'sim.'. In Rank Aggregation, we use rankings derived from the previously mentioned surrogate metrics as the basis for aggregation and adopt 6 aggregation approaches introduced in Goswami *et al.* [35].

Pretraining-based Model Selection: We follow the preprocessing approach outlined by Sylligardos *et al.* [86] and segment each time series into non-overlapping subsequences of length $l = 1024$ to ensure both fair comparison and computational efficiency. Regarding the feature set, MetaOD [102] employs a suite of 200 features, including statistical and landmark features, which are based on the results of 4 anomaly detectors. To maintain fair and consistent comparison, for methods not specifying their feature set, we utilize Catch22 [40], a selection of 22 univariate time series features, which are not correlated with each other and are fast to compute. We evaluate each method under both *in-distribution (ID)* and *out-of-distribution (OOD)* scenarios. In the ID scenario, the model selector is trained on the entire label set and then applied to the evaluation set. Conversely, in the OOD scenario, we generate 18 different sub-label sets, each excluding one of the 18 datasets. For instance, to evaluate the OOD performance of a model selector for the ECG dataset [63], we utilize a sub-label set comprising the remaining 17 datasets, excluding the ECG dataset itself. Our goal is to encompass the primary categories of pretraining-based methods as outlined in our taxonomy (Figure 3). To achieve this, we re-implement the pipelines for the CLF [86, 99] and RG [44, 95] methods, where we use a random forest [15] to construct model selectors for both methods to ensure a fair comparison. For UReg and CFact, two meta-recommenders proposed by Navarro *et al.* [67] for known and novel data respectively, we evaluate both methods in ID and OOD scenarios following the official pipeline. A direct comparison of the three solutions [70, 84, 105] in kNN is not feasible due to the unavailability of their source code. Therefore, following the practice in Navarro *et al.* [67], we implement a Nearest Neighbor baseline as a proxy. For MetaOD, we follow the official training pipeline and retrain the model on the label set. Following ISAC [45], we re-implement the algorithm which begins by clustering the training data using G-means [37] and then selecting the best-performing model for a new dataset based on its nearest cluster.

Model Generation: Methods in this category generally rely on anomaly scores produced by all detectors in the candidate model set, except for the Booster [97], which aims to enhance the performance of any given anomaly detector. In our evaluation, we use the anomaly score generated from the Global Best model as input to the Booster to assess its effectiveness. For OE, anomaly scores are standardized to Z-values before ensemble aggregation following Aggarwal *et al.* [5]. In HITS, we use the aggregated authority score as the anomaly score for the time series. For Aug [20, 41], we first implement a Majority Voting baseline to evaluate the effectiveness of learning-based detector pruning. This Majority Voting approach encompasses solely the preliminary phase of discovering a reliable initial set, thereafter utilizing the aggregated labels to train an anomaly classifier. 'Orig' refers to the original implementation, while 'Ens' indicates the average ensemble of reliable anomaly detectors as inferred by this method. In Clean [20], four variants are considered in terms of how we obtain the initial training data. 'Majority' uses initial labels identified as anomalies by consensus among 25% of detectors. 'Individual' aggregates the top 5% anomalies detected by each detector. 'Ratio' sums all anomaly scores and selects 15% with the highest scores. 'Avg' calculates the average anomaly score and then sets a threshold to determine initial labels.

4.4 Evaluation Measures

Statistical Validation: To conduct a thorough statistical analysis of the performance of automated solutions, we employ the Wilcoxon test [91] for pairwise comparison across various datasets. Additionally, for comparing multiple solutions across multiple datasets, we use the Friedman test [31] followed by the post-hoc Nemenyi test [68] to determine the ranking among the automated solutions.

Accuracy Evaluation: Aligning with previous anomaly detection evaluation [73, 81], we treat the threshold setting as an orthogonal problem to our primary focus. Consequently, we adopt threshold-independent evaluation metrics that summarize the model performance across all potential thresholds. Typically, the following two metrics commonly employed in classification can be used for

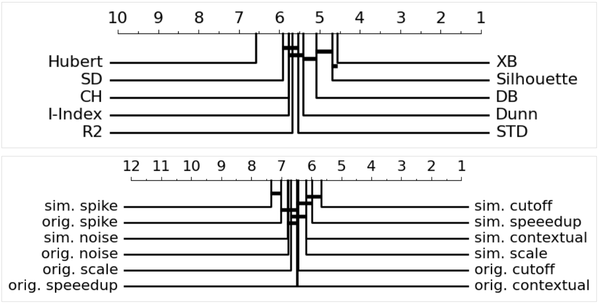


Figure 5: Ranking of variants of (a) CQ and (b) Synthetic by VUS-PR, averaged across the evaluation set. The connecting line shows if the differences are significant or not.

evaluating anomaly detection. *AUC-ROC* [28] quantifies model performance by measuring the area under a curve that plots the true positive rate against the false positive rate, while *AUC-PR* [24] focuses on the relationship between recall and precision. In the case of anomaly detection, the number of true negatives tends to be substantially larger than the number of false positives. This disparity often results in a low false positive rate across various thresholds, rendering only a fraction of the ROC curve relevant. Therefore, AUC-PR is advocated as a more informative metric for imbalanced datasets [59]. Following the common practice of previous work, we use AUC-PR as the evaluation measure in the performance matrix for pretraining-based model selection (see details in Section 3.1.2).

The above two measures are primarily designed for point-based anomalies, treating each point independently and assigning equal weight to the detection of each point in calculating the overall AUC. However, these may not be ideal for assessing subsequence anomaly cases. To address this, the Volume Under the Surface (VUS) [72], which is an extension of the ROC and PR curves, has been introduced for the time series domain. VUS introduces a buffer region at the outliers' boundaries, thereby accommodating the false tolerance of labeling in the ground truth and assigning higher anomaly scores near the outlier boundaries. We integrate both metrics in the VUS family, which are *VUS-ROC* and *VUS-PR*, to ensure a thorough and nuanced evaluation of automated solutions.

Efficiency Evaluation: In addition to the accuracy evaluation of these solutions, we measure the *detection time* during the test phase. It refers to the duration required to obtain a detection result (i.e., the anomaly score) for a given time series. In the process of model selection, the detection time is divided into two components: *execution time*, which measures the time needed to identify the selected model from a given time series, and *detector runtime*, which is the time required for the chosen model to compute and produce the anomaly score. For pretraining-based model selection, our runtime analysis excludes the training phase and instead emphasizes the detection time to offer a targeted evaluation of operational efficiency. For model generation, where the method's output is directly the detection result, the execution time represents the detection time.

5 EXPERIMENTAL RESULTS

In this section, we first assess the effectiveness of different variants within each solution, selecting the best-performing variant for ease of subsequent analysis (Section 5.1). Subsequently, we provide a comprehensive assessment of the accuracy of automated solutions across 18 datasets (Section 5.2). Following this, we investigate the performance of these solutions across various types of anomalies, including point versus sequence anomalies, single versus multiple anomalies, and the influence of anomaly ratio (Section 5.3). We then delve into the model selection, with a specific focus on out-of-distribution experiments and the distribution of model selections (Section 5.4). Lastly, in addition to the accuracy results, we present an evaluation of the detection time for these solutions (Section 5.5).

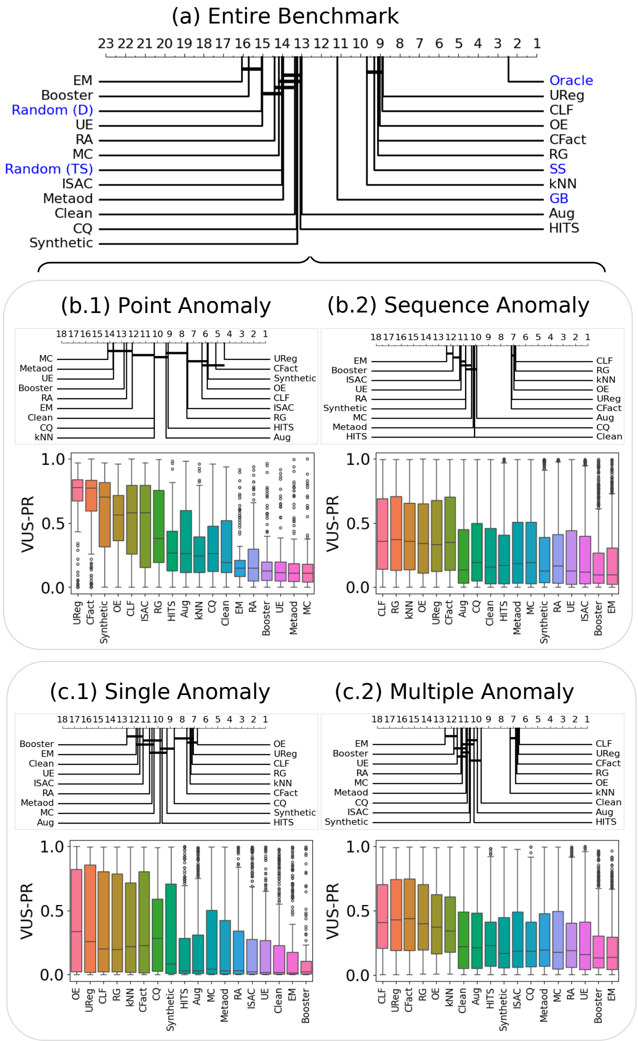
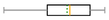
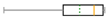
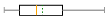
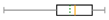
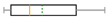
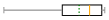
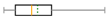
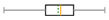
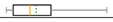
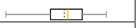
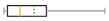
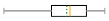
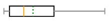
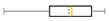
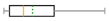
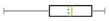
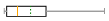
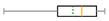
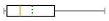
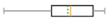
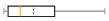
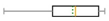
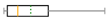
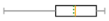
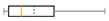
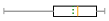
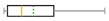
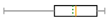
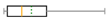
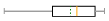
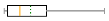
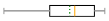
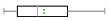
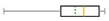
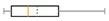
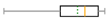
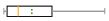
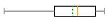
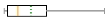
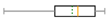
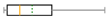
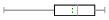
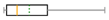
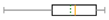
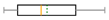
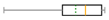
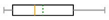
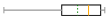
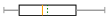
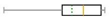
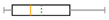
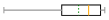
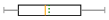
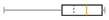
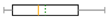
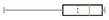
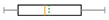
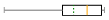
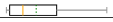
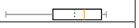
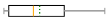
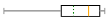
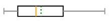
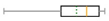
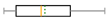
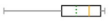
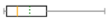
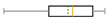
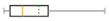
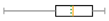
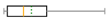
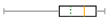
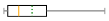
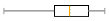
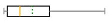
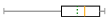
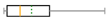
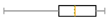
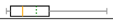
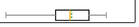


Figure 6: Accuracy Evaluation of 18 solutions (top-performing variant for each method) and 5 baselines across (a) the benchmark. (b.1) and (b.2) depict the accuracy evaluation on time series that contain point or sequence anomaly. (c.1) and (c.2) depict the accuracy evaluation on time series that contain single or multiple anomalies, respectively.

5.1 Best Variant for Each Method

We begin by determining the most effective variant for each method included in our benchmark. To facilitate comparison, we opt for the variant that best represents each method for subsequent evaluation.

Table 4: Summary of accuracy evaluation for baseline and automated solutions with boxplots showing score distributions for VUS-PR and VUS-ROC metrics (mean highlighted in green and median in orange). In the Rank column, boldface indicates a method surpassing SS, while italics indicate a method performing worse than Random (TS).

	Method	Rank	VUS-PR	VUS-ROC
Baseline	Oracle	**1**		
	SS	7		
	GB	14		
	Random (TS)	*26*		
	Random (D)	*38*		
Model Selection	EM	*40*		
	CQ (XB)	19		
	CQ (Silhouette)	21		
	MC (1N)	*32*		
	MC (5N)	*28*		
	Synthetic (sim. cutoff)	18		
	Synthetic (orig. cutoff)	*29*		
	Synthetic (sim. speedup)	22		
	Synthetic (sim. contextual)	24		
	RA (Borda)	*31*		
	RA (Trimmed Borda)	*33*		
	kNN (ID)	8		
	kNN (OOD)	15		
	ISAC (ID)	*27*		
	ISAC (OOD)	*30*		
	MetaOD (ID)	25		
	MetaOD (OOD)	*35*		
	RG (ID)	**5**		
	RG (OOD)	12		
	CLF (ID)	**2**		
	CLF (OOD)	13		
	UReg (ID)	**3**		
	UReg (OOD)	11		
	CFact (ID)	**6**		
	CFact (OOD)	23		
Model Generation	OE (Avg)	9		
	OE (Max)	10		
	OE (AOM)	**4**		
	UE	*37*		
	Booster	*39*		
	HITS	17		
	Aug (Orig)	*34*		
	Aug (Ens)	16		
	Clean (Majority)	20		
	Clean (Individual)	*36*		

We report statistical significant results with a 95% confidence level in the Wilcoxon test to establish the rankings of the variants within each method. In Figure 5, we provide an example of rankings of variants within the CQ and Synthetic categories. For CQ, the variant that utilizes the Xie-Beni index emerges as the top-performing one, a result consistent with the findings of Ma *et al.* [60]. For Synthetic, approaches involving synthetic anomaly removal generally outperform those that inject anomalies directly into the original time series. Among the anomaly injection techniques, *cutoff*, *speedup*, and *contextual* exhibit superior performance. Apart from these two methods, in the case of MC, no significant statistical difference is observed among its three variants. In terms of the RA method, *Borda* and *Trimmed Borda* emerge as the most effective. Within the model generation category, for OE, *AOE* demonstrates superior performance over *Max* and *Avg*. For Clean, the *Majority* outperforms the other three variants, while for Aug, the ensemble of scores from reliable anomaly detectors (i.e., *Ens*) surpasses other approaches (i.e., *Majority Voting* and *Orig*), which rely on the probability output by classifiers of a point being an outlier as the anomaly score.

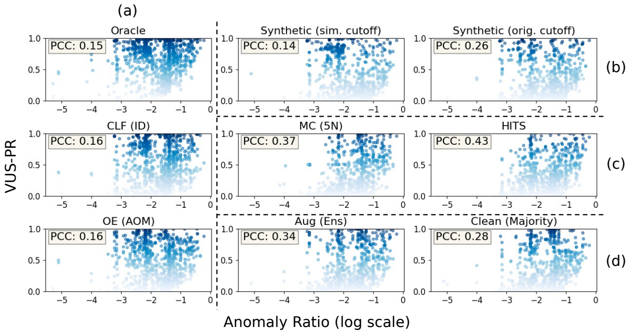


Figure 7: Influence of anomaly ratio on performance (VUS-PR). PCC indicates the Pearson Correlation Coefficient.

5.2 Benchmark Accuracy Evaluation

In Figure 6(a), we present a ranking of 5 baselines and top-performing variants from 18 methods, with the baseline highlighted in blue. The rankings form 4 clusters, with Oracle representing the theoretical upper bound of model selection. The second cluster includes the pretraining-based model selection methods and OE. The methods that perform worse than GB form the fourth cluster. In Table 4, we select 35 representative variants out of 60 in the benchmark to compare against 5 baselines. Upon examining the results, it is evident that pretraining-based model selectors under ID scenarios perform optimally. In fully unsupervised contexts, OE (AOM) emerges as the top performer. When comparing pretraining-based methods under ID with SS, which is the expected lower bound for in-distribution pretraining-based model selection, some methods such as kNN, ISAC, and MetaOD, fail to surpass this baseline. Additionally, no pretraining-based methods in OOD scenarios exceed the performance of OOE, highlighting the importance of addressing performance drops in OOD situations. Among surrogate metrics, CQ and Synthetic are identified as the most effective criteria. The MC does not achieve promising results; however, by extending the concept of centrality calculation in one shot to a recursive manner like in HITS, the ranking sees a significant improvement of 10 positions. Furthermore, out of the 35 variants listed in the table, ranging from ISAC (OOD) to EM, 14 of them were unable to outperform random choice. When considering all 60 variants, a total of 41 variants failed to outperform random choice.

5.3 Analysis on Anomaly Types

In this section, we proceed to evaluate the performance of automated solutions across different types of anomalies.

5.3.1 *Point and Sequence Anomaly.*

We compare the performance of different methods on time series data with point-based anomalies in Figure 6(b.1) and sequence-based anomalies in Figure 6(b.2). UReg emerges as the leading approach for point anomalies. In this scenario, Synthetic works well and even outperforms OE and other model selection methods. When it comes to sequence anomalies, there are two distinct clusters in the rankings. OE and Pretraining-based model selection methods exhibit superior performance compared to other solutions. CLF performs the best, followed by the RG. However, in the case of sequence anomalies, the performance of Synthetic is not as impressive as it is for point anomalies. The best method under a fully unsupervised setting becomes OE.

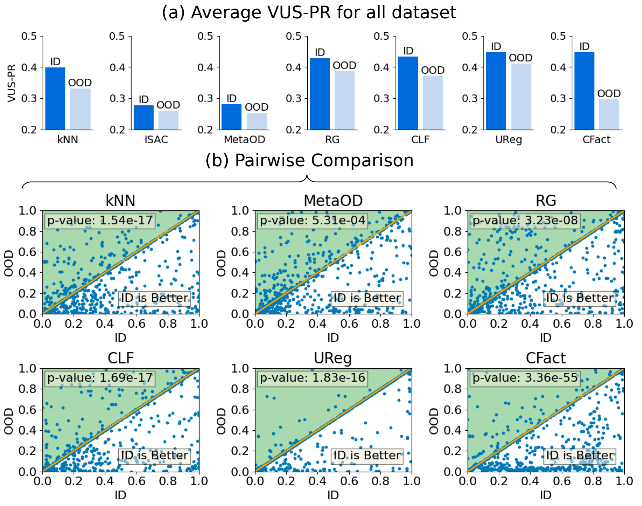


Figure 8: OOD experiments results. (a) depicts the average VUS-PR and (b) provides a pairwise comparison with p-values determined by the one-sided Wilcoxon signed rank test.

5.3.2 Single and Multiple Anomaly. We analyze the performance differences between single anomaly scenarios as depicted in Figure 6(c.1), where the time series contains only one anomaly and multiple anomaly scenarios in Figure 6(c.2), characterized by the presence of several anomalies in the time series. It is observed that no model selection techniques are able to outperform OE in terms of the single anomaly scenario. The most effective unsupervised surrogate metric in this context is CQ, followed by Synthetic. On the other hand, in the scenario of multiple anomalies, the pretraining-based model selectors exhibit greater advantages. The optimal model generation approach is OE, followed by the Clean strategy.

5.3.3 Influence of Anomaly Ratio. In Figure 7, we investigate the influence of anomaly ratio. The figure is segmented into four groups. Within Group (a), consisting of the three top-performing methods, it is observed that the higher the anomaly ratio, the detection algorithm tends to perform better. Both OE (AOM) and CLF (ID) demonstrate correlation coefficients that closely mirror those of the Oracle, indicating their effectiveness across a range of anomaly ratios. In Group (2), the comparison between Synthetic (sim. cutoff) and Synthetic (orig. cutoff) reveals that the initial anomaly removal step helps alleviate the impact of false negatives inherent in the synthetic anomaly injection approach, therefore reducing the influence of anomaly ratio. The anomaly ratio has a notable influence on two centrality-based methods (i.e., MC and HITS) in Group (c). For Aug and Clean in Group (d), the higher correlation suggests that a larger number of anomalies enables the method to acquire high-quality labels while reducing the influence of noisy labels.

5.4 Analysis on Model Selection

In this section, we delve into model selection. We first analyze the impact of OOD on pretraining-based model selectors, then investigate the model selected distribution compared to Oracle.

5.4.1 Out-of-distribution Experiments. To evaluate the performance of pretraining-based model selectors in scenarios where the test time series is dissimilar to any of those used in training data, we examine their effectiveness under OOD conditions. For this purpose, model selection algorithms are trained on all but one dataset (see details in Section 4.3). Based on the results shown in Figure 8(a), it can be observed that all methods exhibit lower performance in OOD compared to ID scenarios. Specifically, RG, CLF, UReg, and CFact demonstrate similar performance levels within the ID setting. However, regression-based techniques such as RG and UReg outperform other methods in OOD scenarios. Statistical differences between ID and OOD are present in 6 out of the 7 methods analyzed. A more detailed pairwise comparison of these 6 methods is provided in Figure 8(b). These findings highlight two key insights: (i) Existing pretraining-based model selection algorithms suffer from significant performance degradation when dealing with out-of-distribution cases. (ii) While the Classification-based method is effective in ID situations, its performance drops greatly in OOD cases, suggesting a potential for overfitting. In contrast, regression-based methods (UReg and Regression) aim to minimize the mean squared error, thereby enabling the model selector to predict not just the optimal model but also its expected performance. This optimization approach proves to be more robust to novel data.

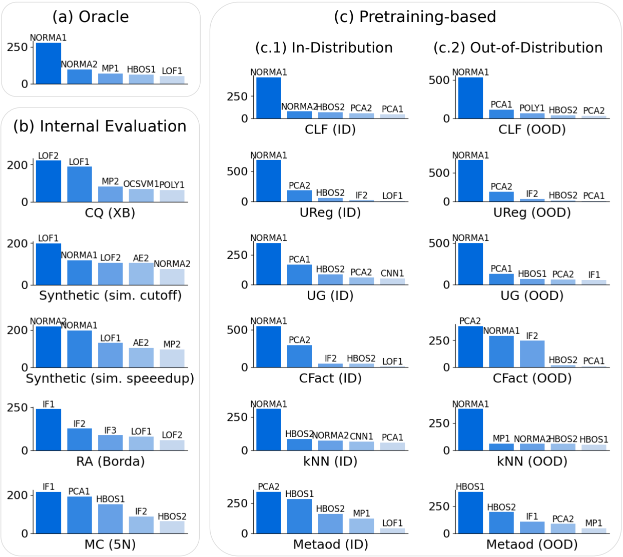


Figure 9: Overview of model selected distribution. The identification of each candidate model refers to Table 2.

5.4.2 Model Selected Distribution. We examine the distribution of models selected by each model selector. Figure 9(a) depicts the frequency of selection for the top 5 models in Oracle. There is an overlap in the top 5 models chosen across all model selectors. However, the top choice of models from internal evaluation methods differs from Oracle's top selection, which is NORMA1. The distribution of models chosen by pretraining-based model selectors exhibits notable differences between ID and OOD scenarios. The similarity in model selection by UReg in both ID and OOD scenarios suggests greater robustness in OOD conditions, a conclusion supported by earlier findings in Section 5.4.1. However, UReg significantly overestimates NORMA1, with more than half of its model predictions favoring NORMA1. Conversely, internal evaluation methods such

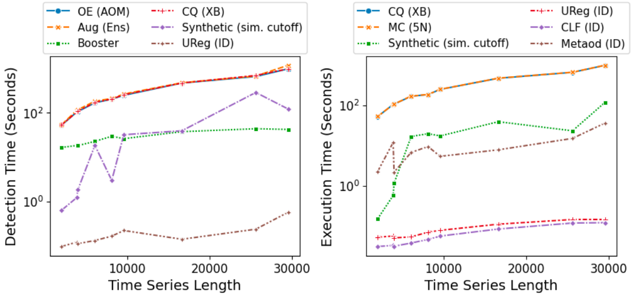


Figure 10: Detection time (left) and execution time (right) comparison of different automated solutions.

as CQ (XB), RA (Borda), and MC (5N) do not include NORMA1 among their top 5 selections, showing a preference for LOF and IF anomaly scores. This preference might indicate that internal evaluations may lean towards specific models' anomaly scores rather than accurately identifying the optimal model for anomaly detection.

5.5 Detection Time Analysis

We now discuss the detection time for automated solutions. In Figure 10, we present the detection time for automated anomaly detection solutions and the execution time for model selection approaches. Notably, methods that rely on obtaining anomaly scores from all candidate models, such as Outlier Ens, Label Aug, Clustering, and Model Centrality, exhibit significantly higher detection time. During testing, pretraining-based anomaly detectors demonstrate notably low detection and execution times. The primary difference lies in the process of feature extraction. Among these methods, MetaOD is particularly computation-intensive as it necessitates running certain anomaly detectors to gather landmark features. This is achieved by downsampling the time series by a factor of ten when computing surrogate evaluation measures on simulated time series. Otherwise, the computational demand for this solution would increase several times over.

6 DISCUSSION AND FUTURE RESEARCH

By now, we have gained a comprehensive understanding of the effectiveness and efficiency of each method. Figure 11 offers an illustration of the balance between these two factors. The choice of automated solutions is contingent upon the specific use cases. For instances with access to historical labeled data, the pretraining-based model selector is advisable due to its accuracy and quick processing times. Conversely, if the scenario lacks any reference data, Outlier Ens emerges as the optimal choice. Moreover, if there is prior knowledge regarding the anomaly types present in the test data, Figure 6 can be referred to obtain a comprehensive ranking of various solutions for different anomaly types. For example, in cases expected to contain point anomalies, Synthetic Anomaly Injection could surpass Outlier Ens as the preferred choice in unsupervised contexts. Despite these insights, it is worth noting that the research attention in this field remains insufficient, with numerous promising avenues yet to be explored. Through our benchmark, we identify some research opportunities as follows.

Domain Generalization: As highlighted in Section 5.4.1, the performance gap between in-distribution and out-of-distribution cases in the context of pretraining-based model selectors represents a significant barrier to their broader adoption. Although largely underexplored in the field of anomaly detection model selection, the concept of domain generalization, aimed at learning a model that can generalize to an unseen test domain, has received extensive attention and achieved great progress over the years [90]. This body of work lays a solid groundwork for integrating domain generalization strategies into pretraining-based model selection processes.

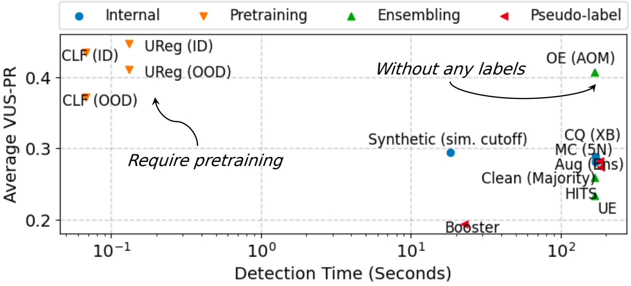


Figure 11: Comparison of average VUS-PR versus detection time for time series of 6,000 data points. The analysis includes four groups of methods, each denoted by unique markers.

Explore Time Series Traits: Many of the existing automated solutions are proposed for tabular datasets and fail to consider the unique attributes of time series data. On one hand, leveraging the characteristics of time series involves specialized feature extraction techniques. It is important to note that, at present, there is no feature set explicitly tailored for recommending methods for detecting anomalies in time series. With the ongoing advancements in time series representation learning [92, 101], exploring data-driven methods for extracting features also presents a promising avenue. On the other hand, many solutions treat each time step or window in isolation, which may not be optimal for time series analysis. Acknowledging and utilizing the temporal dependencies inherent in time series data is crucial for developing more efficient solutions.

Incremental Automated Solutions: Few works have been proposed for evolving data streams. However, being able to perform automated anomaly detection in streaming data and incrementally update the selection or generation algorithm offers significant advantages for both academic research and industrial applications.

7 CONCLUSION

In this study, our focus is on the automated identification of the optimal anomaly detector for an unlabelled time series. A noticeable gap exists in this area, as current methods are evaluated on different datasets and assumptions, without specific focus on the time series domain. To shed light on the current research status of this challenge, we conduct a thorough review and establish a taxonomy of works in this field. Additionally, we provide an end-to-end benchmark that includes 18 methods across the primary categories of existing research and make it publicly available to facilitate progress in tackling this challenge. Furthermore, we perform a systematic evaluation of these automated solutions on 1918 different time series across 18 different datasets. Our findings reveal that 60% of the proposed methods do not exhibit superior performance compared to random choice. This study highlights the critical importance and ongoing demand for automated solutions within the time series domain, acting as a call for further research on this topic.

REFERENCES

[1] [n.d.]. http://iops.ai/dataset_detail/?id=10.

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 265–283.

[3] Charu C. Aggarwal. 2017. *Outlier Analysis* (2 ed.). Springer International Publishing. https://doi.org/10.1007/978-3-319-47578-3

[4] Charu C Aggarwal and Charu C Aggarwal. 2017. *An introduction to outlier analysis*. Springer.

[5] Charu C Aggarwal and Saket Sathe. 2015. Theoretical foundations and algorithms for outlier ensembles. *Acm sigkdd explorations newsletter* 17, 1 (2015), 24–47.

[6] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. 2017. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing* 262 (2017), 134–147. https://doi.org/10.1016/j.neucom.2017.04.070

[7] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2623–2631.

[8] Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M. Hausdorff, Nir Giladi, and Gerhard Troster. 2010. Wearable Assistant for Parkinson's Disease Patients With the Freezing of Gait Symptom. *IEEE Transactions on Information Technology in Biomedicine* 14, 2 (2010), 436–446. https://doi.org/10.1109/TITB.2009.2036165

[9] Maroua Bahri, Flavia Salutari, Andrian Putina, and Mauro Sozio. 2022. AutoML: state of the art with a focus on anomaly detection, challenges, and research directions. *International Journal of Data Science and Analytics* 14, 2 (2022), 113–126.

[10] Pawel Benecki, Szymon Piechaczek, Daniel Kostrzewa, and Jakub Nalepa. 2021. Detecting Anomalies in Spacecraft Telemetry Using Evolutionary Thresholding and LSTMs. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Lille, France) *(GECCO '21)*. Association for Computing Machinery, New York, NY, USA, 143–144. https://doi.org/10.1145/3449726.3459411

[11] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J Christopher Westland. 2011. Data mining for credit card fraud: A comparative study. *Decision support systems* 50, 3 (2011), 602–613.

[12] Paul Boniol, Michele Linardi, Federico Roncallo, Themis Palpanas, Mohammed Meftah, and Emmanuel Remy. 2021. Unsupervised and scalable subsequence anomaly detection in large data series. *The VLDB Journal* (2021), 1–23.

[13] Paul Boniol, Michele Linardi, Federico Roncallo, Themis Palpanas, Mohammed Meftah, and Emmanuel Remy. 2021. Unsupervised and scalable subsequence anomaly detection in large data series. *The VLDB Journal* 30, 6 (2021).

[14] J-C de Borda. 1781. Mémoire sur les élections au scrutin: Histoire de l'Académie Royale des Sciences. *Paris, France* 12 (1781).

[15] Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.

[16] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 93–104.

[17] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. *ACM SIGMOD Record* 29, 2 (May 2000), 93–104. https://doi.org/10.1145/335191.335388

[18] Guilherme O Campos, Arthur Zimek, Jörg Sander, Ricardo JGB Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E Houle. 2016. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data mining and knowledge discovery* 30 (2016), 891–927.

[19] Luis M. Candanedo and Véronique Feldheim. 2016. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. *Energy and Buildings* 112 (2016), 28–39. https://doi.org/10.1016/j.enbuild.2015.11.071

[20] Lei Cao, Yizhou Yan, Yu Wang, Samuel Madden, and Elke A Rundensteiner. 2023. AutoOD: Automatic Outlier Detection. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–27.

[21] Sourav Chatterjee, Rohan Bopardikar, Marius Guerard, Uttam Thakore, and Xiaodong Jiang. 2022. MOSPAT: AutoML based Model Selection and Parameter Tuning for Time Series Anomaly Detection. *arXiv preprint arXiv:2205.11755* (2022).

[22] Stéphan Clémençon and Jérémie Jakubowicz. 2013. Scoring anomalies: a M-estimation formulation. In *Artificial Intelligence and Statistics*. PMLR, 659–667.

[23] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. 1990. STL: A seasonal-trend decomposition. *J. Off. Stat* 6, 1 (1990), 3–73.

[24] Jesse Davis and Mark Goadrich. 2006. The Relationship between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning* (Pittsburgh, Pennsylvania, USA) *(ICML '06)*. Association for Computing Machinery, New York, NY, USA, 233–240. https://doi.org/10.1145/1143844.1143874

[25] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. 2020. A survey on ensemble learning. *Frontiers of Computer Science* 14 (2020), 241–258.

[26] Sunny Duan, Loic Matthey, Andre Saraiva, Nicholas Watters, Christopher P Burgess, Alexander Lerchner, and Irina Higgins. 2019. Unsupervised model selection for variational disentangled representation learning. *arXiv preprint arXiv:1905.12614* (2019).

[27] Philippe Esling and Carlos Agon. 2012. Time-series data mining. *ACM Computing Surveys (CSUR)* 45, 1 (2012), 1–34.

[28] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27, 8 (2006), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010 ROC Analysis in Pattern Recognition.

[29] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. 2019. Auto-sklearn: efficient and robust automated machine learning. *Automated machine learning* 2019 (2019), 113–134.

[30] Pavel Filonov, Andrey Lavrentyev, and Artem Vorontsov. 2016. Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model. arXiv:1612.06676 [cs.LG]

[31] Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Amer. Statist. Assoc.* 32 (1937), 675–701.

[32] Nicolas Goix. 2016. How to evaluate the quality of unsupervised anomaly detection algorithms? *arXiv preprint arXiv:1607.01152* (2016).

[33] Nicolas Goix, Anne Sabourin, and Stéphan Clémençon. 2015. On anomaly ranking and excess-mass curves. In *Artificial Intelligence and Statistics*. PMLR, 287–295.

[34] Markus Goldstein and Andreas Dengel. 2012. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track* 9 (2012).

[35] Mononito Goswami, Cristian Challu, Laurent Callot, Lenon Minorics, and Andrey Kan. 2022. Unsupervised model selection for time-series anomaly detection. *arXiv preprint arXiv:2210.01078* (2022).

[36] Scott David Greenwald. 1990. *Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information.* Thesis. Massachusetts Institute of Technology. https://dspace.mit.edu/handle/1721.1/29206 Accepted: 2005-10-07T20:45:22Z.

[37] Greg Hamerly and Charles Elkan. 2003. Learning the k in k-means. *Advances in neural information processing systems* 16 (2003).

[38] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. 2022. Adbench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems* 35 (2022), 32142–32159.

[39] Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems* 212 (2021), 106622.

[40] Trent Henderson and Ben D Fulcher. 2021. An empirical evaluation of time-series feature sets. In *2021 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 1032–1038.

[41] Dennis Hofmann, Peter VanNostrand, Huayi Zhang, Yizhou Yan, Lei Cao, Samuel Madden, and Elke Rundensteiner. 2022. A demonstration of AutoOD: a self-tuning anomaly detection system. *Proceedings of the VLDB Endowment* 15, 12 (2022), 3706–3709.

[42] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. 2019. *Automated machine learning: methods, systems, challenges*. Springer Nature.

[43] Alexander Ihler, Jon Hutchins, and Padhraic Smyth. 2006. Adaptive Event Detection with Time-Varying Poisson Processes. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Philadelphia, PA, USA) *(KDD '06)*. Association for Computing Machinery, New York, NY, USA, 207–216. https://doi.org/10.1145/1150402.1150428

[44] Minqi Jiang, Chaochuan Hou, Ao Zheng, Songqiao Han, Hailiang Huang, Qingsong Wen, Xiyang Hu, and Yue Zhao. 2024. ADGym: Design Choices for Deep Anomaly Detection. *Advances in Neural Information Processing Systems* 36 (2024).

[45] Serdar Kadioglu, Yuri Malitsky, Meinolf Sellmann, and Kevin Tierney. 2010. ISAC–instance-specific algorithm configuration. In *ECAI 2010*. IOS Press, 751–756.

[46] E. Keogh, T. Dutta Roy, U. Naik, and A Agrawal. [n.d.]. Multi-dataset Time-Series Anomaly Detection Competition 2021, https://compete.hexagon-ml.com/practice/competition/39/.

[47] Jon M Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46, 5 (1999), 604–632.

[48] Anna Korba, Stephan Clémençon, and Eric Sibony. 2017. A learning theory of ranking aggregation. In *Artificial Intelligence and Statistics*. PMLR, 1001–1010.

[49] Kwei-Herng Lai, Daochen Zha, Guanchu Wang, Junjie Xu, Yue Zhao, Devesh Kumar, Yile Chen, Purav Zumkhawaka, Minyang Wan, Diego Martinez, et al. 2021. Tods: An automated time series outlier detection system. In *Proceedings of the aaai conference on artificial intelligence*, Vol. 35. 16060–16062.

[50] N. Laptev, S. Amizadeh, and Y. Billawala. 2015. *S5 - A Labeled Anomaly Detection Dataset, version 1.0(16M)*. https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70

[51] Nikolay Laptev, Saeed Amizadeh, and Ian Flint. 2015. Generic and scalable framework for automated time-series anomaly detection. In *Proceedings of the*

21th ACM SIGKDD international conference on knowledge discovery and data mining. 1939–1947.

[52] Yuening Li, Zhengzhang Chen, Daochen Zha, Kaixiong Zhou, Haifeng Jin, Haifeng Chen, and Xia Hu. 2021. Autood: Neural architecture search for outlier detection. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2117–2122.

[53] Yuening Li, Daochen Zha, Praveen Venugopal, Na Zou, and Xia Hu. 2020. Pyodds: An end-to-end outlier detection system with automated machine learning. In *Companion Proceedings of the Web Conference 2020*. 153–157.

[54] Zhi Li, Hong Ma, and Yongbing Mei. 2007. A Unifying Method for Outlier and Change Detection from Data Streams Based on Local Polynomial Fitting. In *Advances in Knowledge Discovery and Data Mining (Lecture Notes in Computer Science)*, Zhi-Hua Zhou, Hang Li, and Qiang Yang (Eds.). Springer, Berlin, Heidelberg, 150–161. https://doi.org/10.1007/978-3-540-71701-0_17

[55] Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. 2013. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications* 36, 1 (2013), 16–24.

[56] Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. 2020. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *international conference on machine learning*. PMLR, 6127–6139.

[57] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining*. IEEE, 413–422.

[58] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. 2013. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM international conference on data mining*. SIAM, 252–260.

[59] Jorge M Lobo, Alberto Jiménez-Valverde, and Raimundo Real. 2008. AUC: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography* 17, 2 (2008), 145–151.

[60] Martin Q Ma, Yue Zhao, Xiaorong Zhang, and Leman Akoglu. 2023. The need for unsupervised outlier model selection: A review and evaluation of internal evaluation strategies. *ACM SIGKDD Explorations Newsletter* 25, 1 (2023).

[61] Pankaj Malhotra, L. Vig, Gautam M. Shroff, and Puneet Agarwal. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. In *ESANN*.

[62] Henrique O Marques, Ricardo JGB Campello, Arthur Zimek, and Jörg Sander. 2015. On the internal evaluation of unsupervised outlier detection. In *Proceedings of the 27th international conference on scientific and statistical database management*. 1–12.

[63] G.B. Moody and R.G. Mark. 2001. The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine* 20, 3 (2001), 45–50. https://doi.org/10.1109/51.932724

[64] George B Moody and Roger G Mark. 1992. MIT-BIH Arrhythmia Database. https://doi.org/10.13026/C2F305

[65] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. 2018. DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. *Ieee Access* 7 (2018), 1991–2005.

[66] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed. 2019. DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series. *IEEE Access* 7 (2019), 1991–2005. https://doi.org/10.1109/ACCESS.2018.2886457

[67] Jose Manuel Navarro, Alexis Huet, and Dario Rossi. 2023. Meta-Learning for Fast Model Recommendation in Unsupervised Multivariate Time Series Anomaly Detection. In *AutoML Conference 2023*.

[68] Peter Nemenyi. 1963. *Distribution-free Multiple Comparisons*. Ph.D. Dissertation. Princeton University.

[69] Thanh Trung Nguyen, Uy Quang Nguyen, et al. 2016. An evaluation method for unsupervised anomaly detection algorithms. *Journal of Computer Science and Cybernetics* 32, 3 (2016), 259–272.

[70] Mladen Nikolić, Filip Marić, and Predrag Janičić. 2013. Simple algorithm portfolio for SAT. *Artificial Intelligence Review* 40, 4 (2013), 457–465.

[71] Ioannis Paparrizos. 2018. *Fast, scalable, and accurate algorithms for time-series analysis*. Ph.D. Dissertation. Columbia University.

[72] John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S. Tsay, Aaron Elmore, and Michael J Franklin. 2022. Volume Under the Surface: A New Accuracy Evaluation Measure for Time-Series Anomaly Detection. *Technical Report LIPADE-TR-N7, Université Paris Cité* (2022).

[73] John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S Tsay, Themis Palpanas, and Michael J Franklin. 2022. TSB-UAD: an end-to-end benchmark suite for univariate time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 8 (2022), 1697–1711.

[74] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[75] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[76] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and José del R. Millàn. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*. 233–240. https://doi.org/10.1109/INSS.2010.5573462

[77] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

[78] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*. 4–11.

[79] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis* (Gold Coast, Australia QLD, Australia) *(MLSDA'14)*. Association for Computing Machinery, New York, NY, USA, 4–11. https://doi.org/10.1145/2689746.2689747

[80] Meyer Scetbon, Gabriel Peyré, and Marco Cuturi. 2022. Linear-time gromov wasserstein distances using low rank couplings and costs. In *International Conference on Machine Learning*. PMLR, 19347–19365.

[81] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment* 15, 9 (2022), 1779–1797.

[82] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. 1999. Support vector method for novelty detection. In *Proceedings of the 12th International Conference on Neural Information Processing Systems (NIPS'99)*. MIT Press, Cambridge, MA, USA, 582–588.

[83] Yanyao Shen and Sujay Sanghavi. 2019. Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning*. PMLR, 5739–5748.

[84] Prabhant Singh and Joaquin Vanschoren. 2022. Meta-Learning for Unsupervised Outlier Detection with Optimal Transport. *arXiv preprint arXiv:2211.00372* (2022).

[85] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2828–2837. https://doi.org/10.1145/3292500.3330672

[86] Emmanouil Sylligardos, Paul Boniol, John Paparrizos, Panos Trahanias, and Themis Palpanas. 2023. Choose wisely: An extensive evaluation of model selection for anomaly detection in time series. *Proceedings of the VLDB Endowment* 16, 11 (2023), 3418–3432.

[87] Markus Thill, Wolfgang Konen, and Thomas Bäck. 2020. *MarkusThill/MGAB: The Mackey-Glass Anomaly Benchmark, https://doi.org/10.5281/zenodo.3762385*. https://doi.org/10.5281/zenodo.3762385

[88] Chris Thornton, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. 2013. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 847–855.

[89] Alexander von Birgelen and Oliver Niggemann. 2018. *Anomaly Detection and Localization for Cyber-Physical Production Systems with Self-Organizing Maps*. Springer Berlin Heidelberg, Berlin, Heidelberg, 55–71. https://doi.org/10.1007/978-3-662-57805-6_4

[90] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. 2022. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[91] Frank Wilcoxon. 1992. Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*. Springer, 196–202.

[92] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2022. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations*.

[93] Xuanli Lisa Xie and Gerardo Beni. 1991. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 13, 08 (1991), 841–847.

[94] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 world wide web conference*. 187–196.

[95] Lin Xu, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. 2008. SATzilla: portfolio-based algorithm selection for SAT. *Journal of artificial intelligence research* 32 (2008), 565–606.

[96] Yuan Yao, Abhishek Sharma, Leana Golubchik, and Ramesh Govindan. 2010. Online anomaly detection for sensor systems: A simple and efficient approach. *Performance Evaluation* 67, 11 (2010), 1059–1075. https://doi.org/10.1016/j.peva.2010.08.018 Performance 2010.

[97] Hangting Ye, Zhining Liu, Xinyi Shen, Wei Cao, Shun Zheng, Xiaofan Gui, Huishuai Zhang, Yi Chang, and Jiang Bian. 2023. UADB: Unsupervised Anomaly Detection Booster. *arXiv preprint arXiv:2306.01997* (2023).

[98] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Zachary Zimmerman, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2018. Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. *Data Mining and Knowledge Discovery* 32, 1 (Jan. 2018), 83–123. https://doi.org/10.1007/s10618-017-0519-9

[99] Yuanxiang Ying, Juanyong Duan, Chunlei Wang, Yujing Wang, Congrui Huang, and Bixiong Xu. 2020. Automated model selection for time-series anomaly detection. *arXiv preprint arXiv:2009.04395* (2020).

[100] Jaemin Yoo, Yue Zhao, Lingxiao Zhao, and Leman Akoglu. 2023. DSV: an alignment validation loss for self-supervised outlier model selection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 254–269.

[101] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8980–8987.

[102] Daochen Zha, Kwei-Herng Lai, Mingyang Wan, and Xia Hu. 2020. Meta-AAD: Active anomaly detection with deep reinforcement learning. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 771–780.

[103] Yue Zhao, Zain Nasrullah, Maciej K Hryniewicki, and Zheng Li. 2019. LSCP: Locally selective combination in parallel outlier ensembles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 585–593.

[104] Yue Zhao, Ryan Rossi, and Leman Akoglu. 2021. Automatic unsupervised outlier model selection. *Advances in Neural Information Processing Systems* 34 (2021), 4489–4502.

[105] Yue Zhao, Sean Zhang, and Leman Akoglu. 2022. Toward unsupervised outlier model selection. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 773–782.

[106] Arthur Zimek, Ricardo JGB Campello, and Jörg Sander. 2014. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *Acm Sigkdd Explorations Newsletter* 15, 1 (2014), 11–22.