TSB-AutoAD: Towards Automated Solutions for Time-Series Anomaly Detection

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

Despite decades of research on time-series anomaly detection, the effectiveness of existing anomaly detectors remains constrained to specific domains - a model that performs well on one dataset may fail on another. Consequently, developing *automated* solutions for anomaly detection remains a pressing challenge. However, the AutoML community has predominantly focused on supervised learning solutions, which are impractical for anomaly detection due to the lack of labeled data and the absence of a well-defined objective function for model evaluation. While recent studies have evaluated standalone anomaly detectors, no study has ever evaluated automated solutions for selecting or generating scores in an automated manner. In this study, we (i) provide a systematic review and taxonomy of automated solutions for time-series anomaly detection, categorizing them into selection, ensembling, and generation methods; (ii) introduce TSB-AutoAD, a comprehensive benchmark encompassing 20 standalone methods and 70 variants; and (iii) conduct the most extensive evaluation in this area to date. Our benchmark includes state-of-the-art methods across all three categories, evaluated on TSB-AD, a recently curated heterogeneous testbed from nine domains. Our findings reveal a significant gap, where over half of the existing solutions do not statistically outperform a simple random choice. Foundation models that claim to offer generalized, one-size-fits-all solutions have yet to deliver on this promise. While naive ensembling achieves high accuracy, it comes at a substantial computational overhead. Conversely, methods leveraging historical datasets enable fast inference but suffer under out-of-distribution conditions. To address this trade-off, we propose a selective ensembling solution, which combines model selection with ensembling to offer a lightweight, practical balance between accuracy and efficiency. We open-source TSB-AutoAD and highlight the need for more robust and efficient solutions.

PVLDB Reference Format:
Qinghua Liu, Seunghak Lee, and John Paparrizos. TSB-AutoAD: Towards Automated Solutions for Time-Series Anomaly Detection. PVLDB, 18(11): 4364 - 4379, 2025.
doi:10.14778/3749646.3749699

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

Proceedings of the VLDB Endowment, Vol. 18, No. 11 ISSN 2150-8097.
doi:10.14778/3749646.3749699

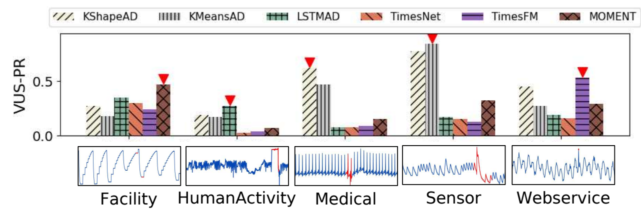

Figure 1: Detection accuracy (VUS-PR) for six representative anomaly detectors across five domains in the TSB-AD benchmark [83]. The red triangle indicates the model with the best detection accuracy: different winners for each domain, supporting the need for automated solutions.

1 INTRODUCTION

Advances in sensing, networking, storage, and processing technologies have enabled the large-scale collection of data, including time series [59, 64, 65, 69, 76, 77, 84, 106]. Time-series analysis has emerged as a field of significant interest, offering critical insights into a wide range of phenomena [49, 89, 97, 111, 112]. A wide array of time-series mining tasks, including clustering [15, 48, 102, 103, 109, 110, 114], classification [39, 40, 101], similarity search [37, 41, 100, 105, 107, 108, 113, 151], and anomaly detection [22–25, 27, 83], has been explored in the literature. *Time-series anomaly detection*, which describes the process of analyzing a time series to identify abnormal patterns, has become critical across multiple scientific fields and industries [81]. 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 [16], network intrusion detection [74], and webservice monitoring [146].

Motivation. The detection of anomalies in time series has received ample academic and industrial attention for over six decades [46, 96]. This interdisciplinary interest spans from data mining and databases to the machine learning community, evolving from traditional statistical methods to neural networks and, lately, foundation models [17, 98]. However, as depicted in Figure 1, our study, along with other recent benchmark studies [83, 104, 127], reveals that no single stand-alone anomaly detector universally outperforms others across different domains. The issue of the absence of a one-size-fits-all model persists, even with the advent of foundation models [83, 135]. Despite the vast amount of anomaly detection models, a critical question remains: *How can we automate time-series anomaly detection by selecting, ensembling, or generating models?* Achieving optimal performance requires in-depth domain knowledge, data distribution, and a comprehensive understanding of the myriad of methods. This necessity drives data analysts into an exhaustive, computationally expensive, costly, and time-consuming

Table 1: Comparison of existing studies for automated anomaly detection with TSB-AutoAD which provides the most comprehensive testbed, covering a wide range of base algorithms-statistical (Stat), neural network-based (NN), and foundation models (FM). It also encompasses a broad spectrum of automated solutions, including meta-learning-based (Meta), internal evaluation (Internal), ensembling-based (Ensemble), and generation-based methods (Generation).

Benchmark	Time Series	Base AD Algorithms				# Automated Solutions			
		#	Stat	NN	FM	Meta	Internal	Ensembling	Generation
Ma *et al.* [86]	✗	8	✓	✗	✗	0	5	2	0
Goswami *et al.* [54]	✓	5	✓	✓	✗	0	4	0	0
Sylligardos *et al.* [134]	✓	12	✓	✓	✗	1	0	1	0
TSB-AutoAD (Ours)	✓	**40**	✓	✓	✓	**7**	**5**	**5**	**3**

trial-and-error process. Consequently, developing automated solutions for time-series anomaly detection is of paramount importance.

Despite numerous efforts made to investigate automated anomaly detection solutions, as outlined in Table 1, these studies exhibit several limitations. These include (i) insufficient evaluation of automated solutions, where previous studies omit entire categories of automated solutions, which limits a comprehensive understanding of the field; (ii) restricted diversity of base anomaly detection (AD) algorithms, as most studies rely on a narrow selection of base algorithms and exclude the latest foundation models, thereby constraining evaluations across different algorithmic landscapes; (iii) the use of different datasets across different studies, which pose substantial challenges for conducting a meta-analysis of their empirical performance. Automation and anomaly detection have been recognized as grand challenges across multiple sectors [2, 3], emphasizing the need to critically assess the current state of the field and whether it has been an illusion of progress. Given these limitations and the critical role of automated solutions in time-series anomaly detection, it is essential to conduct a comprehensive study to thoroughly assess the advancements in this field.

Challenges. Automated anomaly detection is notoriously challenging, primarily due to the intrinsic difficulties of obtaining sufficient labeled data along with its inherently unsupervised nature [20, 104]. This scarcity of labeled data (i.e., inliers and outliers) hinders the accurate comparison of different models, limiting effective model validation and selection [13, 95]. For instance, in a given time series, it is nearly impossible to predefine a validation set with known inlier and outlier labels for model comparison. Moreover, the absence of a universal objective function further complicates automation in anomaly detection. Automated processes evaluate model performance using well-defined quality metrics, such as accuracy for classification [121] or deviation from actual values for forecasting [9], but anomaly detection lacks a standardized evaluation criterion. Additionally, time series exhibit unique characteristics, such as temporal dependencies, varied sampling rates, and continuous values, that differ significantly from those in tabular or image data. This disparity makes automated solutions originally designed for other data types less effective in the context of time series.

Furthermore, conducting a systematic study presents substantial challenges due to the dispersion of proposed automated solutions, which are scattered across various communities such as machine learning [91, 154], data mining [5, 86], and data management [31, 134]. These challenges arise from the difficulties associated with locating, integrating, and implementing these methods

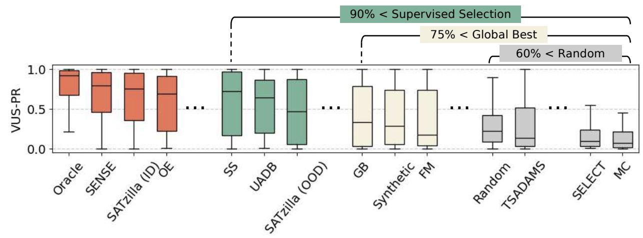

Figure 2: Accuracy overview of automated time-series anomaly detection solutions, ranked by VUS-PR from left (highest) to right. Methods are grouped into clusters, with ratios indicating the number of methods per cluster.

into a unified framework to investigate the performance variance of different design choices. Moreover, these methods operate under a range of assumptions, from reliance on historical data to entirely unsupervised approaches. This variability complicates the determination of the most effective method under different application scenarios, subsequently impeding the broader adoption and practical application of these methodologies.

Contribution. To address these challenges and assess the current research landscape, we introduce TSB-AutoAD and present the most comprehensive study to date, including a taxonomy and systematic review of automated anomaly detection methods. We evaluate 20 solutions with 70 variants across diverse time-series domains, measuring effectiveness, runtime, robustness to distribution shifts, and performance across anomaly types and candidate sets. Based on our findings, we propose Selective Ensembling (SENSE), which combines the strengths of model selection and ensembling to enhance robustness and generalization, while maintaining runtime efficiency. SENSE is designed as a modular plug-in framework that allows the integration of a selector and an ensembling strategy, chosen from multiple candidates evaluated in this study.

Results. As shown in Figure 2, we present a performance overview of automated solutions evaluated on TSB-AutoAD. Our study reveals a significant gap in automated anomaly detection solutions, with over *half* of the evaluated variants failing to outperform random selection and 75% underperforming a naive globally best model (GB) strategy. Only 10% of the methods are able to outperform Supervised Selection (SS), the common practice of labeling a subset of data and using the best-performing model for the rest. Among the methods, OE, which ensembles anomaly scores from all candidates, demonstrates strong robustness but at a high computational cost. Meanwhile, automated solutions that leverage historical datasets suffer performance degradation on out-of-distribution (OOD) time series. SENSE improves accuracy while maintaining runtime efficiency, offering a practical trade-off.

We start with a discussion of the problem statement and related works (Section 2). Then, we present our contributions:

- We formulate a taxonomy for automated solutions for time-series anomaly detection, and review relevant works (Section 3).
- We introduce TSB-AutoAD benchmark to facilitate the exploration of the performance of automated solutions (Section 4).
- We conduct a comprehensive and rigorous evaluation of 20 automated solutions with 70 variants across nine time-series domains and provide research insights (Section 5).
- We summarize findings and outline future research (Section 6).

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

2 PRELIMINARY

We first provide the problem statement for automated solutions (Section 2.1), followed by a discussion of related works (Section 2.2).

2.1 Problem Statement

Definition. We denote the time-series signal observed from N sensors over time T as $X = \{x_1, ..., x_T\}$ with each $x_t \in \mathbb{R}^N$. Anomaly detection involves applying an anomaly detector M to X to generate an anomaly score series $S = \{s_1, ..., s_T\}$ for each time step, where $s_t \in \mathbb{R}$ and a higher score indicates a greater likelihood of an anomaly. Selecting and configuring the anomaly detector M usually requires the intervention of a human expert. Therefore in this study, we define *automated solution* as the task to automatically generate the anomaly score S from a set of candidate models $C = \{\mathrm{M}_1, \mathrm{M}_2, ..., \mathrm{M}_n\}$ without the need for human intervention.

Terminology. It is important to distinguish between the *base AD algorithm* and the *candidate model set*. The base AD algorithm refers to the different detection algorithms ((e.g., LOF [29])), each with multiple variants defined by hyperparameters. Each variant, with its specific hyperparameter settings (e.g., LOF with the number of neighbors set to 20), constitutes a candidate model. Therefore, automated solutions operate on the candidate model set.

Scenario. This automated process typically occurs through one of three approaches: (i) selecting a single model from the candidate set C, (ii) aggregating predictions from multiple models in C through ensembling, or (iii) generating a new mode M_{New} derived from the candidate models in C. Additionally, automated solutions can be classified based on their level of supervision required, operating either in a fully unsupervised manner or leveraging knowledge from historical datasets via meta-learning.

It is important to distinguish these approaches from Bayesian Optimization (BO) [129], where models or hyperparameters are optimized based on known ground truth and a predefined objective function (e.g., minimizing prediction error in supervised learning). In anomaly detection, however, it is typically infeasible to obtain labeled instances of anomalies and normal data for the given test time series ahead of time. Moreover, there is no universal objective function for anomaly detection tasks, which limits the applicability of BO [13]. While methods that utilize historically labeled datasets also require supervision, they differ from BO in that, during inference, they do not require labeled samples as BO does.

2.2 Related Work

2.2.1 Time-Series Anomaly Detection. We begin with the definitions of anomaly and then introduce different anomaly detectors.

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 [20]. Point anomalies are individual data points that significantly deviate from the majority of the data. Contextual 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.

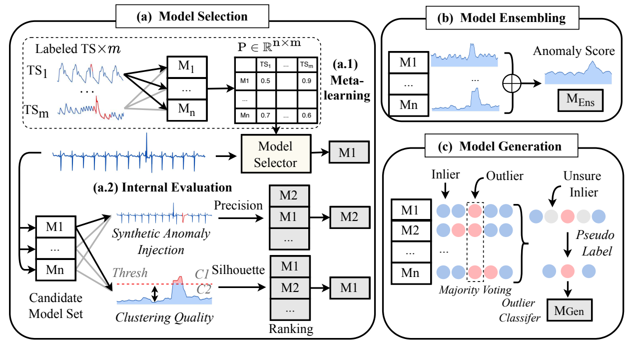

Figure 3: An overview of TSB-AutoAD benchmark. We use M_1, M_2, and M_n to represent the candidate models.

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 [19, 29]; (ii) density-based methods, identify anomalies by focusing on isolated behaviors within the overall data distribution, rather than measuring nearest-neighbor distances [4, 78]; 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 [90, 125]. In this way, the anomalies are identified by significant deviations between predictions and the actual data.

2.2.2 Automated Machine Learning (AutoML). AutoML offers a promising methodology for developing machine learning systems without human intervention [14, 63]. This approach addresses what is formally recognized as the Combined Algorithm Selection and Hyper-parameter (CASH) problem. Several successful studies have been conducted to tackle this issue [8, 44, 137]. 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 [121]) and deviation from actual data (for forecasting [9]). However, the broader AutoML community has predominantly focused on supervised learning applications, where labeled validation sets are available to facilitate model comparison and hyperparameter optimization [42, 79, 141]. This gap is particularly notable in **unsupervised** anomaly detection, compounded by the lack of unsupervised quality metrics to evaluate anomaly detection algorithms effectively [13].

2.2.3 Automated Anomaly Detection Studies. Several efforts have been made to evaluate automated anomaly detection methods, as illustrated in Table 1. Ma *et al.* [86] assess unsupervised model selection for anomaly detection, yet their investigation predominantly focuses on only one category of methods, namely, internal evaluation strategies, and does not extend to time-series data. Similarly, the work by Goswami *et al.* [54] represents the first effort to address the unsupervised model selection challenge in time-series anomaly detection; however, their methodology is limited to internal evaluation techniques. In contrast, Sylligardos *et al.* [134] focus on meta-learning-based approaches, although their research is primarily centered on model selection through the use of pretrained classifiers. None of the existing studies provide comprehensive coverage across all categories of automated anomaly detection methods, highlighting the need for a more holistic evaluation.

3 AUTOMATED SOLUTIONS FOR TIME-SERIES ANOMALY DETECTION

In Figure 3, we present an overview of the automated solution pipeline in TSB-AutoAD. We will start with the introduction of the proposed taxonomy (Section 3.1) and then elaborate on the details of works from the three different categories: model selection (Section 3.2), model ensembling (Section 3.3), model generation (Section 3.4) in the subsequent sections.

3.1 Taxonomy Development

We present a taxonomy of existing automated solutions for anomaly detection, as illustrated in Figure 4. These approaches can be categorized into three main categories: model selection, model ensembling, and model generation. **Model selection** refers to identifying the best model and its corresponding hyperparameters from the candidate set. Subsequently, the selected model is utilized for anomaly detection. Within the model selection category, the existing literature can be further categorized into two groups: meta-learning-based and internal evaluation methods. The former leverages the knowledge of the performance of various anomaly detectors on historical labeled datasets to enable the automated model selection for new datasets. The latter evaluates the effectiveness of a model by using surrogate metrics for anomaly detection, independent of external data such as ground truth labels for anomalies.

Model ensembling aggregates predictions from multiple candidate models using ensemble strategies to enhance robustness and accuracy. **Model generation** entails the construction of a completely new model based on the candidate set, which can then operate as an anomaly detector to produce scores. We will elaborate on these methodologies in detail in the following.

3.2 Model Selection

The task of model selection refers to identifying the best model and its corresponding hyperparameters from a predefined candidate set. This selected model is then used for anomaly detection. Methods in this category involve the use of historical knowledge (Section 3.2.1) and the development of internal evaluation (Section 3.2.2).

3.2.1 Meta-learning-based Methods. These methods are predicated on the principles of meta-learning [13, 140, 142], which leverage meta-knowledge about model performance to improve the selection of selection by observing how different methods perform on different datasets. Specifically, in the context of anomaly detection, these methods require historical datasets annotated with anomalies, utilizing insights from these datasets to select the most appropriate model for new data. As depicted in Figure 3 (a), a historical dataset with labeled anomalies $X_{train} = \{X_1, ..., X_m\}$ is provided, where m represents the number of time series. Subsequently, a

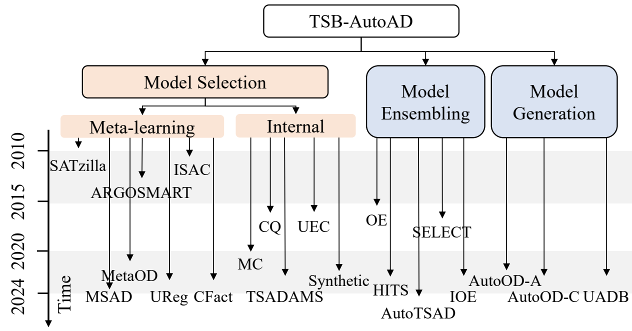

Figure 4: A taxonomy of automated solutions in time-series anomaly detection with chronicle.

performance matrix $P \in \mathbb{R}^{m \times n}$ 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, $P_{i,j}$ corresponds to the i-th anomaly detector's performance on the j-th historical dataset. Given a new data $X_{New} \in \mathbb{R}^{1 \times T}$, 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 applied to the performance matrix, which guides the training of the model selectors, as will be detailed subsequently.

Simple meta-learners identify the best model through straightforward search mechanisms:

(1) ARGOSMART [94] finds the closest (1NN) train data X_i to the given X_{New} based on meta-feature similarity and selects the model with the best performance on X_i dataset.

(2) ISAC [66] clusters meta-train datasets X_{train} based on meta-features. Given X_{New}, it first identifies its closest cluster and selects the best model within this cluster (i.e., the largest average performance on the datasets within this cluster).

Optimiztion-based meta-learners learn task similarity by optimizing performance estimates:

(3) MetaOD [154] is based on *collaborative filtering*: n candidate models are evaluated over m different meta-train datasets, and a matrix factorization process approximates the performance of all models based on a projected matrix of meta-features extracted from the datasets. For a new dataset X_{New}, its meta-features are extracted and then multiplied by the matrix factorization component, yielding a performance prediction for every model in the candidate set.

(4) MSAD [134] converts the model selection process into a *classification* task by training a classifier on X_{train}, each labeled with the best anomaly detector from n candidate models. For a new test time series X_{New}, the model selector classifies it into one of n categories, thereby determining the best model.

(5) SATzilla [148], **(6) UReg [91]**, and **(7) CFact** [91] transform model selection as a *regression* task, utilizing features from labeled datasets to estimate the performance metrics of each anomaly detector. The model selector, functioning as a regressor, is optimized by mean squared error. For the input X_{New}, the model selector predicts the expected performance of each anomaly detector, choosing the one with the highest predicted performance. In contrast to MSAD (i.e., classifier), regression-based methods not only predict which model is recommended but also its expected performance.

3.2.2 Internal Evaluation Methods. These methods evaluate the effectiveness of a model without any reliance on external information (i.e., ground truth labels for anomalies).

Stand-alone evaluation relies solely on each anomaly detector and its corresponding output anomaly score:

(8) Unsupervised Evaluation Curves (UEC) [50] comprise two numerical performance criteria based on Mass-Volume [33] and Excess-Mass [51] curves to compare the performance of anomaly detectors without the need for labeled data. This approach eliminates the reliance on labels for performance evaluation based on Receiver Operating Characteristic (ROC) or Precision-Recall (PR) curves, which typically require labeled data.

(9) Clustering Quality (CQ) [93] utilizes internal validation measures originally designed for clustering algorithms within the context of anomaly detection evaluation. For this, anomaly scores are partitioned into two clusters by setting thresholds (i.e., the abnormal points cluster and the normal points cluster). Subsequently, clustering metrics (e.g., Silhouettes [123]) can be applied to assess their performance, determining the best 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.

Collective evaluation utilizes the interactions among models within the predefined candidate model set:

(10) Model Centrality (MC) [75] 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 [54, 86], 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.

(11) Synthetic Anomaly Injection (Synthetic) is based on the assumption that an effective anomaly detector should exhibit superior performance on data with artificially introduced anomalies [54]. 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 best choice. Chatterjee *et al.* [32] 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 [34] 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 time series instead of the original time series as in Goswam *et al.* [54]. However, while simulated data provides valuable insights, it deviates from real-world scenarios, potentially leading to erroneous decisions.

(12) TSADAMS [54] aggregates imperfect rankings derived from the aforementioned unsupervised surrogate metrics to achieve more reliable rankings of anomaly detectors. Specifically, they explore the application of Kemeny rank aggregation [68], wherein an efficient approximation is implemented through the Borda method [28]. Furthermore, TSADAMS introduces several robust variants of the Borda method, which focus on considering only the top k models and aggregating more reliable rankings.

3.3 Model Ensembling

Ensemble learning integrates the informative knowledge from weak predictive results obtained from various learning algorithms (i.e., different anomaly detectors) to enhance knowledge discovery and predictive performance through adaptive voting schemes [38]. By integrating diverse predictive signals, ensemble methods enhance robustness and mitigate the weaknesses of individual models. These approaches can be broadly classified based on how the ensemble set is constructed: (i) aggregating anomaly scores from all available models without any selection process, or (ii) incorporating a model selection mechanism and ensembling only a subset of models.

(13) Outlier Ensemble (OE) [5] draws an analogy to the bias-variance trade-off in classification and introduces three strategies: *AVG*, which averages scores across detectors to reduce variance; *MAX*, which selects the maximum score per point to reduce bias and highlight outlier-like behavior; and *AOM*, which averages the maximum scores from random detector subsets to balance bias and variance. While AVG provides stability, MAX is more effective in revealing subtle anomalies that may be down-weighted across most detectors, and AOM integrates the strengths of both.

(14) SELECT [119] employs a two-phase ensemble approach that integrates multiple detectors and various consensus techniques to choose ensemble components without supervision. Rather than aggregating predictions from all candidate models, SELECT strategically selects a subset of detector results to assemble through the proposed 'Vertical' and 'Horizontal' selection.

(15) Iterative Outlier Ensemble (IOE) [86] also 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 set. Subsequently, the distance between it and each anomaly score in the candidate set 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.

(16) HITS [86] is adapted in the context of anomaly detection from centrality computation in a network setting [67]. In contrast to Model Centrality (MC), 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.

(17) AutoTSAD [126] is an ensemble system that automatically produces an aggregated anomaly scoring without a need for labeled training data. Specifically, it consecutively executes the three modules: (i) *data generation*, which generates a diverse set of synthetic training time series with injected anomalies, (ii) *algorithm optimization*, which leverages the synthetic training time series to create a pool of optimized algorithm configurations, (iii) *scoring* *ensembling*, which executes the algorithm instances on the test time series, ranks the most effective algorithm instances, and combines their anomaly scores to produce a final anomaly score.

As part of this work, and to address the limitations of naïve ensembling, particularly its high computational cost, we propose SENSE, a Selective Ensembling strategy which integrates model selection (as described in Section 3.2) with ensembling to achieve a practical trade-off between accuracy and runtime efficiency. Comprehensive results and analysis are provided in Section 5.6.

3.4 Model Generation

In contrast to the previous two categories, model generation concentrates on creating an entirely new model tailored to a specific dataset based on the predefined model set. 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 [30]. 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 model and then tuning its parameters, a different approach involves generating pseudo labels to transform the unsupervised problem into a supervised one. This line of research focuses on optimizing the use of existing anomaly detection algorithms, all the while circumventing the need for human-generated labels. The generated pseudo labels, which indicate the likely positions of inliers and outliers, are subsequently used to train a binary classifier that functions as the anomaly detector.

(18) AutoOD-A [31, 60] is built upon the idea that selecting one model from many alternate unsupervised anomaly detectors may not always work well. Instead, it targets combining the best of them. AutoOD-A begins by automatically identifying a small but reliable set of labels (inliers and outliers) 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 [80] to refine the set of reliable objects based on comparisons between logistic regression outcomes and trained outlier classifier until a set of reliable objects does not change.

(19) AutoOD-C [31] starts with a large set of noisy labels and progressively cleans them to produce a more reliable set. The process involves the following 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 [131], 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.

(20) UADB [152] aims 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. The anomaly scores can be refined iteratively.

Table 2: Dataset partitioning for benchmarking automated solutions on TSB-AD-U and TSB-AD-M.

Domain	TSB-AD-U			TSB-AD-M		
	Total	Training	Eval	Total	Training	Eval
WebService	310	220	90	0	0	0
Medical	147	105	42	49	36	13
Facility	143	102	41	50	36	14
Synthetic	122	87	35	0	0	0
HumanActivity	58	41	17	9	6	3
Sensor	44	32	12	78	56	22
Environment	20	14	6	13	9	4
Finance	20	14	6	1	0	1
Traffic	6	4	2	0	0	0
Total	870	619	251	200	143	57

4 TSB-AUTOAD OVERVIEW

In this section, we review the experimental settings of TSB-AutoAD. We begin by providing the setup of the benchmark (Section 4.1) followed by the implementation of automated solutions and baseline methodologies (Section 4.2). Lastly, we discuss the evaluation metrics employed (Section 4.3).

4.1 Experimental Setup

We now introduce the technical platform and implementation, along with the datasets and candidate models we use as follows.

4.1.1 Platform. We conduct our experiments on a server with the following configuration: 2xAMD EPYC 7713 64-Core. The server has two Nvidia A100 GPUs and runs Ubuntu 22.04.3 LTS (64-bit). We implemented the library and scripts that accompany TSB-AutoAD in Python 3.10 with the main following dependencies: Pytorch 1.12 [115] and scikit-learn 1.3.2 [116]. For reproducibility purposes, we open-source the TSB-AutoAD [136], accompanied by a demonstration toolkit for interactive result exploration [82].

4.1.2 Datasets. The issues associated with the quality of time-series anomaly detection datasets, including common flaws such as mislabeling, bias, and feasibility, have significantly hindered progress in evaluation and benchmarking practices [83, 144]. To ensure reliable benchmarking results, we conduct our evaluation of automated solutions using the recently published, heterogeneous, and curated TSB-AD dataset [83]. TSB-AD comprises 870 univariate (TSB-AD-U) and 200 multivariate (TSB-AD-M) time series from nine different domains, including web services [6, 144], medical [52, 56], facility [45, 88], synthetic [70, 71], human activity [12, 122], sensor [7, 62], environment [1, 144], finance [130, 138], and traffic [6]. A detailed dataset description is available in the GitHub [136].

For evaluation, the time series from each domain are partitioned into two subsets, as shown in Table 2: (i) Training set – utilized for supervised selection and provided as meta-training data for meta-learning-based model selectors. (ii) Evaluation set – made available without access to ground-truth anomaly labels and serving as a testbed for assessing different automated solutions. As a result, the training set consists of 762 time series (619 univariate and 143 multivariate), while the evaluation set contains 308 time series (251 univariate and 57 multivariate).

Table 3: Overview of base algorithms in TSB-AutoAD, categorized into statistical methods (Stat), neural network-based approaches (NN), and foundation models (FM), with applicability to univariate (U) and multivariate (M) time series.

Base Algorithm	Category	Dim	Description
(Sub)-MCD [124]	Stat	U&M	Minimum covariance determinant
Sub-OCSVM [128]	Stat	U&M	Support vector method
(Sub)-LOF [29]	Stat	U&M	Identifying density-based local outliers
(Sub)-KNN [117]	Stat	U&M	Distance to its k-th nearest neighbor
KMeansAD [150]	Stat	U&M	Distance to the centroid of assigned cluster
CBLOF [58]	Stat	M	Clluster-based LOF
POLY [72]	Stat	U	Local polynomial fitting
(Sub)-IForest [78]	Stat	U&M	Isolation Forest
(Sub)-HBOS [53]	Stat	U&M	Height of the bin in histogram
KShapeAD [102]	Stat	U	Identify the normal pattern based on the k-Shape clustering
MatrixProfile [153]	Stat	U	Subsequence exhibiting the greatest nearest neighbor distance
(Sub)-PCA [4]	Stat	U&M	Deviation from hyperplane constructed by eigenvectors
RobustPCA	Stat	M	Identify anomalies by recovering the principal matrix
EIF [57]	Stat	M	Extension of the traditional Isolation Forest algorithm
SR [120]	Stat	U	Spectral residual
COPOD [73]	Stat	M	Copula-based parameter-free detection algorithm
Series2Graph [21]	Stat	U&M	Graph-based subsequence anomaly detection
SAND [26]	Stat	U	Streaming subsequence anomaly detection
AutoEncoder [125]	NN	U&M	Reconstruction error through the encoding-decoding
LSTMAD [87]	NN	U&M	Prediction error using LSTM
CNN [90]	NN	U&M	Prediction error using CNN
Donut [146]	NN	U&M	VAE-based method
OmniAnomaly [133]	NN	U&M	Stochastic recurrent neural network
USAD [11]	NN	U&M	Adversely trained autoencoders
AnomalyTransformer [147]	NN	U&M	Anomaly-Attention mechanism
TranAD [139]	NN	U&M	Self-conditioning and adversarial training
TimeNet [143]	NN	U&M	Temporal 2d-variation modeling
FITS [149]	NN	U&M	Interpolation in the frequency domain
OFA [155]	FM	U&M	Finetuning of pre-trained GPT-2 model
Lag-Llama [118]	FM	U	Decoder-only transformer using lags as covariates
Chronos [10]	FM	U	T5 model pretrained on tokenized time series
TimesFM [35]	FM	U	Pretrained decoder-only attention model with input patching
MOMENT [55]	FM	U	Pre-trained T5 encoder from masked time-series modeling

4.1.3 Candidate Model Set. The candidate models serve as the underlying anomaly detectors from which automated solutions can select or generate final predictions. The accuracy of automated solutions is inherently influenced by the selection and quality of these candidate models, as will later be discussed in Section 5.5. However, the primary objective of this study is to evaluate the relative performance of automated solutions rather than optimizing individual anomaly detectors, making our analysis orthogonal to the specific model choices. To ensure completeness and credibility, we adopt the model set from the TSB-AD benchmark [83]—one of the largest and most recent time-series anomaly detection benchmarks—which includes 40 state-of-the-art algorithms spanning both univariate and multivariate settings (see Table 3)

Once a base algorithm is selected, the next step involves configuring its hyperparameters to instantiate candidate models. Given that more than half of the automated solutions require iteratively applying each candidate model during inference, an unlimited number of candidate models is impractical. Moreover, to ensure the reliability of our evaluation and mitigate the risk of poor configurations degrading performance (i.e., "garbage in, garbage out"), we construct a high-quality candidate model set. Specifically, we perform hyperparameter tuning on the training set, selecting the best configuration for each algorithm to prevent suboptimal parameter choices from compromising model performance. The detailed hyperparameter setting is available on our GitHub repository. As a result, our candidate model set consists of 32 models for univariate and 23 models for multivariate time series. This approach enhances the reliability of subsequent comparisons in model selection and generation processes, ensuring that automated solutions are evaluated under fair and consistent conditions. It is important to note that in real-world applications, practitioners can modify the candidate model set as needed. The candidate selection in this study is intended for benchmarking purposes, providing a unified and consistent testbed for comparing different automated solutions.

4.2 Benchmark Implementations

The following section provides the implementation details for baselines and methods within each category of automated solutions.

4.2.1 Baseline. We employ five types of baselines to evaluate the effectiveness of automated solutions. First, **Oracle** represents the theoretical upper bound for model selection, where the best model for a time series is selected based on its ground truth labels. Second, global best (**GB**) selects the model that exhibits the highest overall performance (i.e., highest average ranking) across the entire evaluation set. Third, supervised selection (**SS**) identifies the best model on the label set of each dataset and then uses it for the remaining evaluation set, which represents the common practice of utilizing a portion of labeled data to determine the most accurate model and then applied it to the test dataset. Compared with GB, which selects a single model globally, SS identifies the best model for each domain, resulting in a total of nine selected models for nine domains. Fourth, random choice (**Random**) simulates the model selection process absent of any prior knowledge or expertise, where a model is randomly chosen for each time series and then applied to that. Fifth, foundation models (**FM**) are pre-trained on large-scale datasets, which enhances their generalization and temporal modeling capabilities for time-series analysis tasks. They can function both as standalone base detection algorithms and as benchmarks against which we compare the performance of automated solutions. In this study, we define the performance of the FM category based on the highest-performing foundation models within our candidate model set: TimesFM [35] for univariate time series and OFA [155] for multivariate time series.

4.2.2 Meta-learning-based Methods. As discussed in Section 3.2.1, meta-learning-based approaches generally follow three key steps: (i) extraction of meta-features, (ii) training of meta-learners, and (iii) applying the trained meta-learner for the model recommendation.

Several studies have explored meta-feature extraction for anomaly detection. For instance, Zhao *et al.*[154] employ a set of 200 meta-features, some of which require running four anomaly detection methods (i.e., HBOS, IForest, LODA, and PCA). However, this approach is computationally expensive and was originally designed for tabular data, lacking considerations for temporal structures. More recently, Navarro *et al.*[91] integrated Catch22 [85], a collection of 22 univariate time-series meta-features—capturing properties such as linear and nonlinear autocorrelation, successive differences, value distributions, and fluctuation scaling—selected from an initial pool of over 4000 features based on their effectiveness in time-series tasks. Their approach demonstrated improved performance in model selection for time-series anomaly detection. To ensure a fair comparison across multiple meta-learners while maintaining computational efficiency, we adopt Catch22 as our meta-feature set. Since Catch22 extracts feature from individual time series, we extend its application to multivariate time-series data following the methodology in Navarro *et al.* [91]. Specifically, for each meta-feature, we compute summary statistics—minimum, first quartile, mean, third quartile, and maximum—resulting in a feature representation of 110 values for multivariate time series.

Moreover, we adopt a unified evaluation pipeline for this category of methods, following the framework established in a recent benchmark for meta-learning-based model selection in time-series anomaly detection [134]. We segment each time series into non-overlapping subsequences of length $l = 1024$, applying model selection to each segment. The final model is selected based on the majority vote among the models identified across all segments. Additionally, we assess each method under both in-distribution (**ID**) and out-of-distribution (**OOD**) scenarios to evaluate their generalization capabilities. In the ID scenario, the model selector is trained on the complete training set and subsequently applied to the evaluation set. In contrast, for the OOD scenario, we construct nine different sub-training sets, each excluding one of the nine domains. For instance, to evaluate a model selector's OOD performance in the Medical domain, the selector is trained on data from the remaining eight domains, ensuring that data from the Medical domain is entirely absent from the training set.

4.2.3 Internal Evaluation. The selection of method variants within this category follows the specifications outlined in their respective original publications, with a detailed list of variants provided later in Table 4. For instance, in our evaluation of the Clustering Quality (CQ) measure, we incorporate ten different clustering quality metrics, including the Xie-Beni index [145] and the Silhouette index [123]. In the case of Model Centrality (MC), we set the number of nearest neighbors to 1, 3, 5, etc., when computing the average distance between anomaly scores. For Synthetic-anomaly-injection-based approaches, we categorize methods into two primary groups: those utilizing the original time series and those applying a simulation protocol as described by Chatterjee *et al.* [32]. The selection of anomaly types follows the strategies proposed by Goswami *et al.* [54], incorporating variations such as spikes and speedup anomalies. Methods that operate on the original time series are denoted as 'Orig', while those employing synthetic transformations are labeled as 'STL'. To efficiently train models within our candidate model set, we follow the setup outlined in Goswami *et al.* [54] and subsample all time series exceeding a length of 2560 by a factor of 10. In TSADAMS, we use rankings derived from the aforementioned surrogate metrics as inputs for rank aggregation methods and adopt six aggregation techniques [54].

4.2.4 Model Ensembling. For methods within this category, we utilize their original publicly available implementations to ensure consistency and reproducibility. For example, in the OE method, anomaly scores are standardized to Z-values prior to ensemble aggregation, following the approach outlined by Aggarwal *et al.* [5]. As AutoTSAD [126] is designed specifically for univariate time series and relies on specialized solutions closely integrated with base AD algorithms—using hyperparameters initialized from heuristics in TimeEval [127]. To ensure the integrity of our experiments, we strictly adhere to its original evaluation protocols, employing the system as implemented in its officially released software.

4.2.5 Model Generation. For AutoOD-A, the 'Orig' variant refers to the original implementation, whereas 'Ensemble' extends the original approach by computing the final anomaly score as the average of the outputs from reliable anomaly detectors identified by the method. In AutoOD-C, four variants are considered in terms of how we obtain the initial training data: (i) 'Majority' uses initial labels identified as anomalies by consensus among 25% of detectors. (ii) 'Individual' aggregates the top 5% anomalies detected by each detector. (iii) 'Ratio' sums all anomaly scores and selects 15% with the highest scores. (iv) 'Avg' calculates the average anomaly score and then sets a threshold to determine initial labels. UADB is designed to enhance any given anomaly score. To provide as much benefit to this solution, we employ the ensembled anomaly score as input—even though in practice only one score is used—to assess whether it can improve the performance of the ensembled score.

4.3 Evaluation Measures

Statistical Validation. To validate the statistical difference of performance among *multiple* automated solutions across *multiple* datasets, we apply the Friedman test [47], followed by the post-hoc Nemenyi test [92] at a 95% confidence level. In the Critical Difference (CD) diagram, methods that do not exhibit statistical differences are connected by black lines.

Accuracy Evaluation. Anomaly detection is commonly framed as a binary classification task, where each time step is labeled as normal or abnormal based on a threshold applied to the anomaly score. While thresholding is typically user-defined or estimated via statistical techniques like Peaks Over Threshold (POT)[132], our study focuses on generating accurate anomaly scores rather than optimizing threshold selection. To ensure fair evaluation, we adopt threshold-independent measures that summarize performance across all possible thresholds[83, 104, 127]. However, existing measures suffer from key limitations: bias (e.g., AUC-ROC overestimating the performance [43]), indiscrimination (e.g., Affiliation [61] yielding uniformly high scores), and lack of adaptability to the sequential nature of time series (e.g., AUC-PR and F-score being sensitive to temporal shifts [36]). To address these issues, we adopt VUS-PR [18, 99] as our primary evaluation metric. VUS-PR enhances robustness to lag by incorporating buffer regions near outlier boundaries, reduces bias, and ensures consistent and fair performance assessment. It also serves as the accuracy measure for evaluating meta-learning-based methods in our benchmark.

Efficiency Evaluation. In addition to the accuracy evaluation of these solutions, we measure the **inference 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 by automated solutions. In model selection, the inference time is divided into two components: **selection time**, which measures the time needed to identify the best model from a given time series, and detector runtime, which is the time required for the selected model to compute and produce the anomaly score.

5 BENCHMARK EVALUATION AND ANALYSIS

In this section, we present a rigorous and comprehensive analysis of the performance of automated solutions, aiming to derive research insights with implications for the novel design and application of automated time-series anomaly detection methods. We aim to provide insights into the following research questions (**RQ**):

- **RQ1.** How far we are at achieving automated, robust, and accurate time-series anomaly detection (Section 5.1)?
- **RQ2.** What are the computational implications and scalability characteristics of these automated solutions (Section 5.2)?

Table 4: Accuracy evaluation with boxplots showing score distributions for VUS-PR (mean in green and median in orange). The best variant for each method is marked with ★. The '# of Wins' represents the number in which a given method outperforms the three baselines, SS, GB, and Random.

	Method	Variant	VUS-PR	Rank	# of Wins		
					SS	GB	Random
Baseline	Oracle	-		1	218	270	308
	SS	-		8	0	212	251
	GB	-		19	88	0	178
	Random	-		28	57	130	0
	FM	-		29	50	143	144
Model Selection	SATzilla	ID ★		2	123	218	277
		OOD		17	89	159	201
	ISAC	ID ★		13	100	175	218
		OOD		31	58	115	155
	ARGOSMART	ID ★		5	129	201	260
		OOD		20	75	139	173
	MetaOD	ID ★		25	81	134	172
		OOD		28	80	138	153
	MSAD	ID ★		3	132	209	268
		OOD		16	96	162	204
	UReg	ID ★		7	110	199	257
		OOD		18	91	145	184
	CFact	ID ★		11	95	186	240
		OOD		24	71	136	165
	CQ	XBS		42	51	125	129
		STD		47	47	118	121
		R2		37	56	134	135
		Hubert		53	57	123	109
		CH ★		34	56	134	135
		Silhouette		63	30	91	89
		I-Index		45	59	131	125
		DB		36	66	121	133
		SD		58	51	116	110
		Dunn		38	65	127	137
	UEC	EM ★		33	40	136	140
		MV		64	30	101	99
	MC	3		67	42	98	72
		5 ★		69	40	99	71
		7		70	38	99	69
		9		72	38	97	68
		12		73	36	95	64
	Synthetic	Orig-spikes		41	48	129	134
		STL-spikes		26	67	142	149
		Orig-scale		39	56	127	142
		STL-scale		21	73	142	170
		Orig-noise		52	60	123	128
		STL-noise		32	72	128	143
		Orig-cutoff		44	58	115	138
		STL-cutoff		31	64	127	147
		Orig-contextual		46	58	110	144
		STL-contextual		35	68	116	145
		Orig-speedup ★		22	82	143	169
		STL-speedup		23	67	142	170
	TSADAMS	Borda ★		50	56	125	119
		Kemeny		55	56	126	110
		Trimmed Kemeny		57	52	126	112
		Partial Borda		59	44	119	110
		Trimmed Borda		56	48	123	119
		MIM		60	58	119	110
Ensembling	OE	AVG ★		4	141	213	268
		MAX		12	103	176	246
		AOM		6	126	204	265
	SELECT	Vertical ★		62	52	100	89
		Horizontal		65	45	92	95
	IOE	-		66	51	100	85
	HITS	-		15	91	166	197
Generation	AutoOD-A	Orig		68	38	93	77
		Ensemble ★		10	118	191	240
		Majority		71	32	84	62
	AutoOD-C	Majority ★		40	67	124	131
		Ratio		51	61	116	109
		Average		54	61	115	113
		Individual		61	49	104	83
	UADB	Orig		14	113	174	218
		Mean_C		48	61	119	124
		STD_C		49	61	119	124
		Mean		43	65	124	131
		STD ★		9	125	195	246

- **RQ3.** How robust are these methods under out-of-distribution conditions and across different types of anomalies (Section 5.3)?
- **RQ4.** How does the performance of automated solutions vary across different types of anomalies (Section 5.4)?

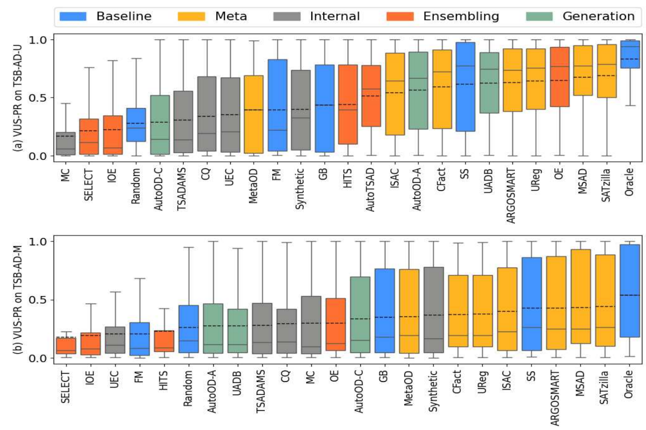

Figure 5: Summary of accuracy evaluation of automated solutions (their best variants) on TSB-AD-U and TSB-AD-M. The methods are arranged from right to left in the boxplot based on the rankings of the average VUS-PR value. The mean is marked by a dashed line and the median by a solid line.

- **RQ5.** How does the choice of candidate model sets affect the overall performance of automated solutions (Section 5.5)?
- **RQ6.** What are the benefits of selective ensembling (Section 5.6)?

5.1 Overall Accuracy Evaluation

In Table 4, we present a comprehensive evaluation of automated solutions across both univariate and multivariate time series, compared against five baselines in terms of both average rankings and the number of "wins" each method achieves over a given baseline across 308 times in total. Unfortunately, the advent of foundation models has not fundamentally transformed the landscape of time-series anomaly detection, nor do they offer a one-size-fits-all solution—a finding consistent with recent studies on foundation models in time-series analysis [83, 135]. Consequently, there still remains a pressing need for robust automated solutions. The top-performing automated solutions appear to be meta-learning-based methods, which leverage knowledge from historical datasets, and ensembling methods, which aggregate wisdom from multiple models. Despite these promising aspects, the overall performance of automated solutions remains below expectations. Among the evaluated variants, only 7 outperform SS (a common practice of using labeled validation data for model selection), and fewer than 20 exceed GB (applying one single model with the most robust performance). Moreover, over *half* of the methods fail to surpass random choice. Although meta-learning-based methods exhibit strong performance in ID scenarios, they experience notable degradation under distribution shifts, as demonstrated by comparisons between ID and OOD cases (further discussed in Section 5.3).

Figure 5 presents an overview of the best-performing variants of automated solutions, distinguishing their performance on univariate and multivariate time series. The overall trend remains consistent, with meta-learning-based methods achieving the highest rankings, followed by OE for univariate data and Synthetic for multivariate time series. OE demonstrates robust performance through a simple score ensembling strategy, underscoring both the potential of ensemble methods and the ongoing need for more efficient and robust automated solutions in time-series anomaly

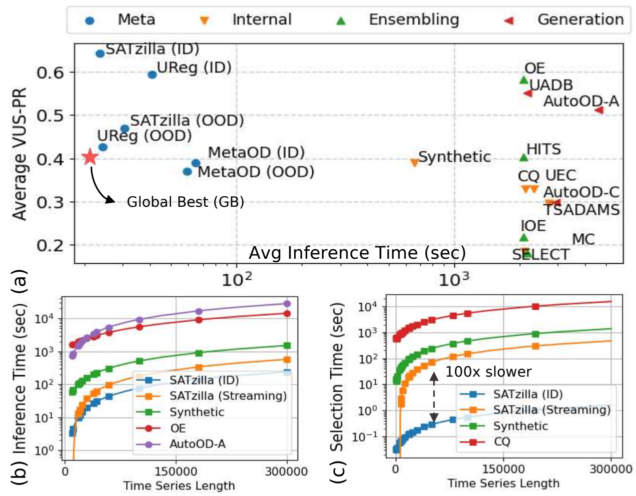

Figure 6: Overview of runtime analysis for automated solutions: (a) illustration of the relationship between VUS-PR and average detection time across the benchmark and the illustration of scalability with respect to (b) inference time and (c) selection time for model selection methods.

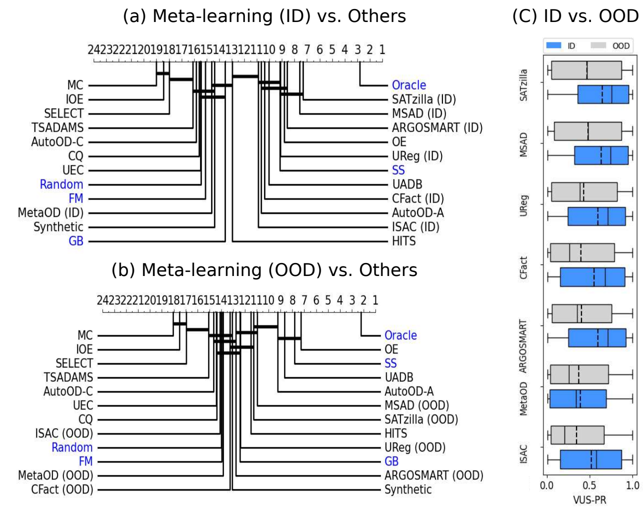

Figure 7: Illustration of the impact of distribution shifts on meta-learning-based methods in (a) ID, (b) OOD cases, and (c) comparison between the two cases.

detection. In contrast, AutoTSAD, which is specifically designed for univariate time series, fails to outperform this simple ensembling approach. UADB, designed to enhance a given anomaly score (specifically, the score produced by OE (Avg) in our study), fails to surpass the performance of OE (Avg) itself, highlighting the need for further advancements in this approach to enhance its effectiveness. Synthetic approach emerges as the most effective unsupervised model selection methodology, particularly excelling in multivariate cases. However, its performance varies depending on the type of synthetic anomalies used. The impact of initial outlier removal is also anomaly-dependent; while it proves ineffective for speedup anomalies, it is beneficial for spike anomalies. These findings underscore the promise of synthetic methodologies while emphasizing the need for more realistic and effective anomaly injection techniques. Finally, internal evaluation measures consistently fail to outperform random selection, underscoring their limited effectiveness in approximating anomaly score performance.

5.2 Runtime Scalability

In Figure 6, we present a runtime analysis of automated solutions. As shown in Figure 6 (a), meta-learning-based methods achieve significantly lower runtimes compared to alternative approaches. This efficiency stems from their ability to select the best model by leveraging historical knowledge instead of performing model selection by iteratively examining each model's performance on the fly. Although these methods experience degradation under distribution shifts, their performance under OOD conditions remains superior to that of most other automated solutions, all while maintaining orders of magnitude lower runtimes. In contrast, OE, which requires the iterative application of each anomaly detector, is computationally expensive despite its robust performance. Moreover, methods such as SELECT and MC are characterized by both slow inference times and lower accuracy. Figures 6 (b) and (c) further demonstrate that the inference time for meta-learning-based methods is considerably lower than that of other model selection techniques. To explore scalability in streaming contexts, despite these methods not being originally designed for such settings, we simulate a scenario where model selection is updated every 300 time points (e.g., equivalent to 5 minutes under one-second sampling intervals). This leads to significantly slower performance compared to static settings, underscoring the need for future adaptation to data streams.

5.3 Out-of-distribution Experiments

To evaluate the performance of meta-learning-based model selectors in scenarios where the test data 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 4.1). Figures 7(a) and (b) show that meta-learning methods drop out of the top three rankings under OOD conditions, while ensembling and generation-based approaches retain stable performance. Nonetheless, meta-learning methods still outperform the Globally Best (GB) baseline in several cases, indicating their potential. Figure 7(c) further highlights that OOD performance degradation is consistent across all meta-learning-based methods. Furthermore, comparisons across different meta-learners reveal that the optimization strategy for model selectors significantly influences performance rankings. Specifically, regression-based (e.g., SATzilla, UReg, CFact) and cross-entropy-based methods (e.g., MSAD) are generally more effective, whereas ranking-based (e.g., MetaOD) and nearest-neighbor-based approaches (e.g., ARGOSMART) perform less favorably.

5.4 Analysis on Anomaly Types

As illustrated in Figure 8, we evaluate the efficacy of various automated solutions (with the best variant selected for each method for clarity) on time series datasets featuring different types of anomalies. While anomaly types are unknown prior to detection, analyzing performance variation across types offers insights into the strengths and limitations of each method. In the case of point anomalies (a), where anomalies occur at individual time steps, the

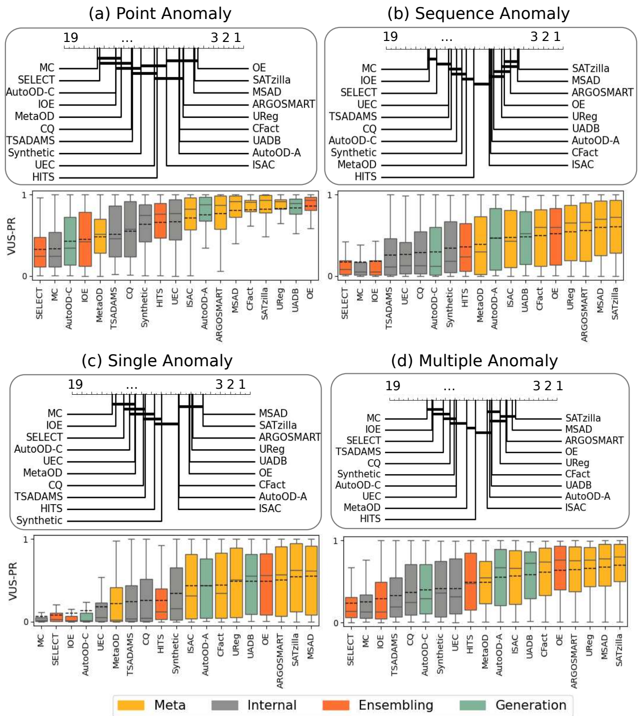

Figure 8: Performance overview under (a) point, (b) sequence, (c) single, and (d) multi anomaly.

ensemble-based method OE demonstrates the highest effectiveness, followed by meta-learning and generation-based approaches such as UADB, with UEC being the most effective internal evaluation method. For sequence anomalies (b), where anomalies span contiguous segments, meta-learning-based methods outperform OE, and the Synthetic approach proves to be the most reliable internal evaluator. We further analyze performance under single-anomaly (c) and multi-anomaly (d) scenarios. In both cases, meta-learning methods and ensembles consistently lead in performance. Synthetic performs particularly well in single-anomaly settings, where the lower contamination ratio reduces false negatives and enhances the effectiveness of synthetic injection. Finally, methods such as MC, SELECT, IOE, and AutoOD-C perform poorly on single-anomaly time series but improve notably when multiple anomalies are present.

5.5 Impact of Different Candidate Sets

In this section, we investigate the impact of candidate model sets by comparing automated solutions using the entire candidate model set versus a subset consisting of the top 10 models from the entire set as identified from the TSB-AD benchmark [83]. Figure 9 (a) presents a pairwise comparison between the entire set and the subset. With a reduced number of available models in the subset, the performance of methods such as SS and top-performing meta-learning-based approaches declines. This reduction is attributed to the restricted selection pool, where the removal of certain models may exclude those that demonstrate higher detection accuracy for specific time series, thereby limiting the effectiveness of model selection. Conversely, a refined subset leads to substantial performance gains for methods that require iterative application of each candidate, such as Synthetic, CQ, and OE—with the most significant

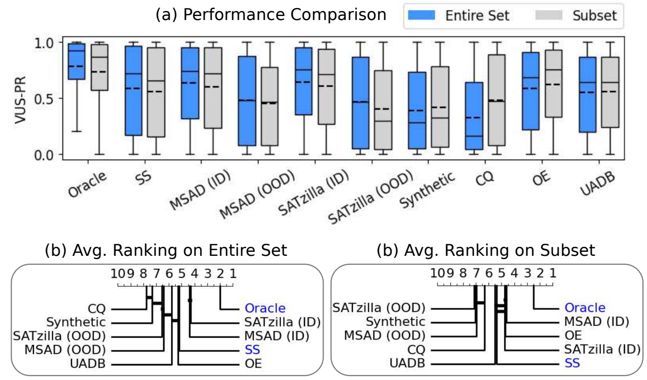

Figure 9: Overview of the impact of candidate model sets. The Entire Set consists of all 40 base AD algorithms, while the Subset includes only the top 10 AD algorithms.

improvement observed in CQ, a relatively weak model selection method. Figures 9 (b) and (c) further illustrate that, although the relative performance for each method differs between the complete set and the subset, the overall ranking of automated solutions remains largely consistent. Meta-learning-based methods and OE remain among the top performers, with OE surpassing the SS baseline when using the refined subset. In contrast, internal evaluation methods still struggle to outperform the SS baseline. However, under out-of-distribution scenarios, they exhibit improved performance compared to meta-learning-based methods when using the refined subset, where meta-learning-based approaches experience performance degradation due to distribution shifts.

5.6 Effectiveness of Selective Ensemble

As discussed previously, relying solely on the top-ranked model introduces risks due to ranking imperfections and model variability, while indiscriminate aggregation of all predictions may lead to unsatisfactory outcomes, where poor models degrade the aggregated results but also consume excessive computational resources. Incorporating the top-k models offers a balance, raising the question of how to combine the strengths of model selection and ensembling while mitigating their respective limitations. To this end, the previously introduced SENSE framework offers a promising solution by aggregating predictions from the top-k ranked models, rather than relying solely on the top-1. As shown in Figure 10, we evaluate the trade-off between accuracy and inference time across different SENSE variants. The benefits of top-k ensembling are particularly evident for the Synthetic variant, where individual model selection performance is less reliable. For SATzilla, performance improvement plateaus after Top-3, suggesting saturation in accuracy and providing an effective balance between performance and computational cost, as inference time increases with k. In OOD settings, achieving comparable performance to OE typically requires ensembling 11–13 models. This suggests that differentiating between ID and OOD cases is a valuable direction for reducing runtime cost.

6 DISCUSSION AND FUTURE RESEARCH

We first present key findings and a practical guide in Section 6.1, followed by a discussion of future research directions in Section 6.2.

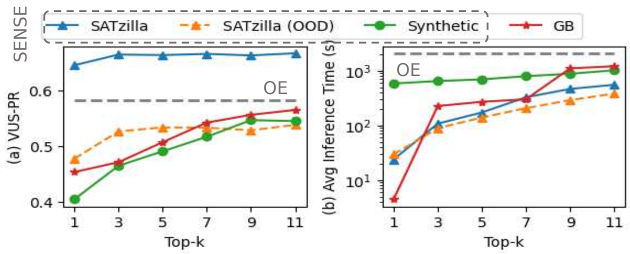

Figure 10: Evaluation of (a) VUS-PR and (b) inference time across different variants of the Selective Ensemble (SENSE) framework under varying values of Top-k. The performance of naive ensembling OE is marked with the gray dotted line.

6.1 Discussion

Position: Current Limitations of Automated Solutions. Despite recent advances in foundation models and anomaly detection algorithms, automated solutions remain more reliable in practice. However, performance disparities persist: only four methods (SATzilla, MSAD, OE, ARGOSMART) surpass Supervised Selection, while 60% of solutions underperform a random baseline and 75% fall below a simple globally best model. Many approaches incur high computational costs by evaluating all candidate models, yet still trail behind meta-learning and simple ensemble strategies. Furthermore, unsupervised surrogate metrics (e.g., MC, UEC, CQ) perform poorly due to limited adaptability to time-series contexts.

Promise: Strengths and Opportunities. (i) The naïve ensemble (OE), which aggregates anomaly scores from all candidate models, delivers surprisingly strong performance, demonstrating the effectiveness of ensembling for bias and variance reduction in anomaly detection. (ii) The optimization strategy within meta-learning-based methods plays a critical role in predictive accuracy. Simple regression- and classification-based losses often yield more robust results, suggesting the need for further exploration of these techniques. (iii) There are substantial differences in runtime across automated solutions. While OE achieves high accuracy, it incurs high computational cost; in contrast, meta-learning approaches are more efficient but less reliable under distribution shift. This highlights the need for methods that balance accuracy and efficiency. (iv) The quality of the candidate model set strongly affects the performance of unsupervised methods. Carefully curated model pools improve reliability and are crucial for real-world deployment. (v) By combining model selection with ensembling, SENSE consistently improves performance while maintaining favorable inference time, offering a balanced solution between accuracy and cost.

Finally, we present a **practical guide** in Figure 11 for selecting automated solutions. When computational resources are sufficient, OE, which ensembles anomaly scores from all candidate models, is recommended and particularly effective for point anomalies. For scenarios with historical labeled data and strict latency requirements, SATzilla and MSAD are preferable, offering strong performance for sequence anomalies. In fully unsupervised settings, the Synthetic anomaly injection approach provides the most reliable model selection and performs well on sequence anomalies. SENSE is introduced as a modular plug-in framework that enables the integration of a model selector and an ensemble component over the top-k candidates. It offers a practical balance between predictive accuracy and computational efficiency, with ensembling as few as three models demonstrating strong performance.

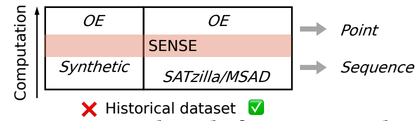

Figure 11: A practical guide for automated solutions.

6.2 Future Research

Despite these insights, it is worth noting that the research attention in this field remains insufficient, with numerous promising avenues yet to be explored. We identify research opportunities as follows.
(1) Domain Generalization. The performance gap between ID and OOD cases in meta-learning-based methods poses a challenge for broader adoption. Despite their advantage in inference efficiency, enhancing domain generalization is critical for improving robustness. SENSE offers an initial attempt to address this issue by ensembling the top-k selected models, but further research is needed to improve selection accuracy under distribution shifts.
(2) Explore Time Series Traits. Many automated solutions are designed for tabular data, overlooking the unique characteristics of time series. Effective automated time-series anomaly detection requires specialized feature extraction techniques. Moreover, many methods treat time steps in isolation, neglecting the temporal dependencies crucial for developing more effective solutions.
(3) Incremental Automated Solutions. Research on automated solutions for evolving data streams remains limited. However, enabling automated detection in streaming settings, along with incremental updates to adapt to concept drift, holds substantial promise for both academic and industrial applications.
(4) Develop Advanced Anomaly Detector. While AutoML aims to reduce manual model design, the development of advanced anomaly detectors remains essential. A more accurate base detector improves overall AutoML performance, and continued progress in detector design directly benefits automated pipelines.

7 CONCLUSION

In this study, we focus on addressing a crucial yet often overlooked research question: *Given a time series, how can we automatically achieve the best anomaly detection performance given a set of candidate models?* However, current methods are proposed from different communities and evaluated on different datasets, without a specific focus on the time series domain. To bridge this gap, we introduce TSB-AutoAD and conduct a comprehensive analysis of automated time-series anomaly detection. Our extensive benchmarking of 20 automated solutions with 70 variants across nine time-series domains, reveals substantial discrepancies: over half of the automated solution variants do not surpass a simple random baseline, yet this analysis also uncovers previously unrecognized but highly effective solutions. This study highlights the critical importance and ongoing demand for automated solutions within the time-series anomaly detection domain, acting as a call for further research on this topic.

ACKNOWLEDGMENTS

All datasets and models were accessed and used solely by The Ohio State University researchers. Additionally, the associated code being released was developed solely by OSU.

REFERENCES

[1] Pacific Marine Environmental Laboratory (PMEL) [n.d.]. *TAO. https://www.pmel.noaa.gov/*. Pacific Marine Environmental Laboratory (PMEL). https://www.pmel.noaa.gov/

[2] 2017. Cyber Attack Detection and Accommodation for Energy Delivery Systems. https://www.energy.gov/sites/prod/files/2017/06/f34/GEGR_ADA_FactSheet_0.pdf.

[3] 2023. Workshop: Solar Applications of Artificial Intelligence and Machine Learning. https://www.energy.gov/eere/solar/workshop-solar-applications-artificial-intelligence-and-machine-learning.

[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.

[7] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P Mathur. 2017. WADI: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd international workshop on cyber-physical systems for smart water networks*. 25–28.

[8] 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.

[9] Ahmad Alsharef, Karan Aggarwal, Sonia, Manoj Kumar, and Ashutosh Mishra. 2022. Review of ML and AutoML solutions to forecast time-series data. *Archives of Computational Methods in Engineering* 29, 7 (2022), 5297–5311.

[10] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815* (2024).

[11] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. 2020. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 3395–3404.

[12] Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M Hausdorff, Nir Giladi, and Gerhard Troster. 2009. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine* 14, 2 (2009), 436–446.

[13] 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.

[14] Rafael Barbudo, Sebastián Ventura, and José Raúl Romero. 2023. Eight years of AutoML: categorisation, review and trends. *Knowledge and Information Systems* 65, 12 (2023), 5097–5149.

[15] Mohini Bariya, Alexandra von Meier, John Paparrizos, and Michael J Franklin. 2021. k-ShapeStream: Probabilistic Streaming Clustering for Electric Grid Events. In *2021 IEEE Madrid PowerTech*. IEEE, 1–6.

[16] 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.

[17] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. 2021. A Review on outlier/Anomaly Detection in Time Series Data. *ACM Computing Surveys (CSUR)* 54, 3 (2021), 1–33.

[18] Paul Boniol, Ashwin K Krishna, Marine Bruel, Qinghua Liu, Mingyi Huang, Themis Palpanas, Ruey S Tsay, Aaron Elmore, Michael J Franklin, and John Paparrizos. 2025. VUS: effective and efficient accuracy measures for time-series anomaly detection. *The VLDB Journal* 34, 3 (2025), 32.

[19] 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.

[20] Paul Boniol, Qinghua Liu, Mingyi Huang, Themis Palpanas, and John Paparrizos. 2024. Dive into time-series anomaly detection: A decade review. *arXiv preprint arXiv:2412.20512* (2024).

[21] Paul Boniol and Themis Palpanas. 2020. Series2Graph: Graph-based Subsequence Anomaly Detection for Time Series. *PVLDB* 13, 11 (2020).

[22] Paul Boniol, John Paparrizos, Yuhao Kang, Themis Palpanas, Ruey S Tsay, Aaron J Elmore, and Michael J Franklin. 2022. Theseus: navigating the labyrinth of time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 12 (2022), 3702–3705.

[23] Paul Boniol, John Paparrizos, and Themis Palpanas. 2023. New Trends in Time Series Anomaly Detection.. In *EDBT*. 847–850.

[24] Paul Boniol, John Paparrizos, and Themis Palpanas. 2024. An interactive dive into time-series anomaly detection. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 5382–5386.

[25] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J Franklin. 2021. SAND in action: subsequence anomaly detection for streams. *Proceedings of the VLDB Endowment* 14, 12 (2021), 2867–2870.

[26] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J Franklin. 2021. SAND: streaming subsequence anomaly detection. *Proceedings of the VLDB Endowment* 14, 10 (2021), 1717–1729.

[27] Paul Boniol, Emmanouil Sylligardos, John Paparrizos, Panos Trahanias, and Themis Palpanas. 2024. Adecimo: Model selection for time series anomaly detection. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 5441–5444.

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

[29] 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.

[30] 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.

[31] 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.

[32] 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).

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

[34] 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.

[35] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*.

[36] 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

[37] Jens E d'Hondt, Haojun Li, Fan Yang, Odysseas Papapetrou, and John Paparrizos. 2025. A Structured Study of Multivariate Time-Series Distance Measures. *Proceedings of the ACM on Management of Data* 3, 3 (2025), 1–29.

[38] 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.

[39] Ziheng Duan, Haoyan Xu, Yueyang Wang, Yida Huang, Anni Ren, Zhongbin Xu, Yizhou Sun, and Wei Wang. 2022. Multivariate time-series classification with hierarchical variational graph pooling. *Neural Networks* 154 (2022), 481–490.

[40] Adam Dziedzic, John Paparrizos, Sanjay Krishnan, Aaron Elmore, and Michael Franklin. 2019. Band-limited training and inference for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 1745–1754.

[41] Jens E d'Hondt, Odysseas Papapetrou, and John Paparrizos. 2024. Beyond the Dimensions: A Structured Evaluation of Multivariate Time Series Distance Measures. In *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 107–112.

[42] Radwa Elshawi, Mohamed Maher, and Sherif Sakr. 2019. Automated machine learning: State-of-the-art and open challenges. *arXiv preprint arXiv:1906.02287* (2019).

[43] 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.

[44] 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.

[45] 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]

[46] Anthony J Fox. 1972. Outliers in time series. *Journal of the Royal Statistical Society: Series B (Methodological)* 34, 3 (1972), 350–363.

[47] Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association* 32, 200 (1937), 675–701.

[48] Apostolos Giannoulidis, Anastasios Gounaris, and John Paparrizos. 2025. BURST: Rendering Clustering Techniques Suitable for Evolving Streams. *Proceedings of the VLDB Endowment* 18, 11 (2025), 4054–4063.

[49] Rahul Goel, Sandeep Soni, Naman Goyal, John Paparrizos, Hanna Wallach, Fernando Diaz, and Jacob Eisenstein. 2016. The social dynamics of language change in online networks. In *International conference on social informatics*. Springer, 41–57.

[50] Nicolas Goix. 2016. How to evaluate the quality of unsupervised anomaly detection algorithms? *arXiv preprint arXiv:1607.01152* (2016).
[51] Nicolas Goix, Anne Sabourin, and Stéphan Clémençon. 2015. On anomaly ranking and excess-mass curves. In *Artificial Intelligence and Statistics*. PMLR, 287–295.
[52] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation* 101, 23 (2000), e215–e220.
[53] 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).
[54] Mononito Goswami, Cristian Challu, Laurent Callot, Lenon Minorics, and Andrey Kan. 2023. Unsupervised Model Selection for Time-series Anomaly Detection. *International Conference on Learning Representations.* (2023).
[55] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. MOMENT: A Family of Open Time-series Foundation Models. In *International Conference on Machine Learning*.
[56] 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.
[57] Sahand Hariri, Matias Carrasco Kind, and Robert J Brunner. 2019. Extended isolation forest. *IEEE transactions on knowledge and data engineering* 33, 4 (2019), 1479–1489.
[58] Zengyou He, Xiaofei Xu, and Shengchun Deng. 2003. Discovering cluster-based local outliers. *Pattern recognition letters* 24, 9-10 (2003), 1641–1650.
[59] Kaisei Hishida, Chunwei Liu, John Paparrizos, and Aaron Elmore. 2025. Beyond Compression: A Comprehensive Evaluation of Lossless Floating-Point Compression. *Proceedings of the VLDB Endowment* 18, 11 (2025), 4396–4409.
[60] 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.
[61] Alexis Huet, Jose Manuel Navarro, and Dario Rossi. 2022. Local evaluation of time series anomaly detection algorithms. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 635–645.
[62] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 387–395.
[63] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. 2019. *Automated machine learning: methods, systems, challenges*. Springer Nature.
[64] Hao Jiang, Chunwei Liu, Qi Jin, John Paparrizos, and Aaron J Elmore. 2020. Pids: attribute decomposition for improved compression and query performance in columnar storage. *Proceedings of the VLDB Endowment* 13, 6 (2020), 925–938.
[65] Hao Jiang, Chunwei Liu, John Paparrizos, Andrew A Chien, Jihong Ma, and Aaron J Elmore. 2021. Good to the Last Bit: Data-Driven Encoding with CodecDB. In *Proceedings of the 2021 International Conference on Management of Data*. 843–856.
[66] Serdar Kadioglu, Yuri Malitsky, Meinolf Sellmann, and Kevin Tierney. 2010. ISAC–instance-specific algorithm configuration. In *ECAI 2010*. IOS Press, 751–756.
[67] Jon M Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46, 5 (1999), 604–632.
[68] Anna Korba, Stephan Clémençon, and Eric Sibony. 2017. A learning theory of ranking aggregation. In *Artificial Intelligence and Statistics*. PMLR, 1001–1010.
[69] Sanjay Krishnan, Aaron J Elmore, Michael Franklin, John Paparrizos, Zechao Shang, Adam Dziedzic, and Rui Liu. 2019. Artificial intelligence in resource-constrained and shared environments. *ACM SIGOPS Operating Systems Review* 53, 1 (2019), 1–6.
[70] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. 2021. Revisiting Time Series Outlier Detection: Definitions and Benchmarks. In *NeurIPS Track on Datasets and Benchmarks*.
[71] 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
[72] 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
[73] Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. 2020. Copod: copula-based outlier detection. In *2020 IEEE international conference on data mining (ICDM)*. IEEE, 1118–1123.
[74] 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.
[75] 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.
[76] Chunwei Liu, Hao Jiang, John Paparrizos, and Aaron J Elmore. 2021. Decomposed bounded floats for fast compression and queries. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2586–2598.
[77] Chunwei Liu, John Paparrizos, and Aaron J Elmore. 2024. AdaEdge: A Dynamic Compression Selection Framework for Resource Constrained Devices. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 1506–1519.
[78] 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.
[79] Hanmo Liu, Shimin Di, Jialiang Wang, Zhili Wang, Jiachuan Wang, Xiaofang Zhou, and Lei Chen. 2025. Structuring Benchmark into Knowledge Graphs to Assist Large Language Models in Retrieving and Designing Models. In *The Thirteenth International Conference on Learning Representations*.
[80] 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.
[81] Qinghua Liu, Paul Boniol, Themis Palpanas, and John Paparrizos. 2024. Time-Series Anomaly Detection: Overview and New Trends. *Proceedings of the VLDB Endowment* 17, 12 (2024), 4229–4232.
[82] Qinghua Liu, Seunghak Lee, and John Paparrizos. 2025. EasyAD: A Demonstration of Automated Solutions for Time-Series Anomaly Detection. *Proceedings of the VLDB Endowment* 18, 12 (2025).
[83] Qinghua Liu and John Paparrizos. 2024. The Elephant in the Room: Towards A Reliable Time-Series Anomaly Detection Benchmark. *Advances in Neural Information Processing Systems* 37 (2024), 108231–108261.
[84] Shinan Liu, Tarun Mangla, Ted Shaowang, Jinjin Zhao, John Paparrizos, Sanjay Krishnan, and Nick Feamster. 2023. AMIR: Active Multimodal Interaction Recognition from Video and Network Traffic in Connected Environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 1 (2023), 1–26.
[85] Carl H Lubba, Sarab S Sethi, Philip Knaute, Simon R Schultz, Ben D Fulcher, and Nick S Jones. 2019. catch22: CAnonical Time-series CHaracteristics: Selected through highly comparative time-series analysis. *Data Mining and Knowledge Discovery* 33, 6 (2019), 1821–1852.
[86] 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).
[87] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. (2015).
[88] Aditya P Mathur and Nils Ole Tippenhauer. 2016. SWaT: A water treatment testbed for research and training on ICS security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE, 31–36.
[89] Kathy McKeown, Hal Daume III, Snigdha Chaturvedi, John Paparrizos, Kapil Thadani, Pablo Barrio, Or Biran, Suvarna Bothe, Michael Collins, Kenneth R Fleischmann, et al. 2016. Predicting the impact of scientific concepts using full-text features. *Journal of the Association for Information Science and Technology* 67, 11 (2016), 2684–2696.
[90] 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.
[91] 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*.
[92] Peter Nemenyi. 1963. *Distribution-free Multiple Comparisons*. Ph.D. Dissertation. Princeton University.
[93] 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.
[94] Mladen Nikolić, Filip Marić, and Predrag Janičić. 2013. Simple algorithm portfolio for SAT. *Artificial Intelligence Review* 40, 4 (2013), 457–465.
[95] Christian O'Leary, Farshad Ghassemi Toosi, and Conor Lynch. 2023. A Review of AutoML Software Tools for Time Series Forecasting and Anomaly Detection. *ICAART (3)* (2023), 421–433.
[96] ES Page. 1957. On problems in which a change in a parameter occurs at an unknown point. *Biometrika* 44, 1/2 (1957), 248–252.
[97] Ioannis Paparrizos. 2018. *Fast, scalable, and accurate algorithms for time-series analysis.* Ph.D. Dissertation. Columbia University.
[98] John Paparrizos, Paul Boniol, Qinghua Liu, and Themis Palpanas. 2025. Advances in Time-Series Anomaly Detection: Algorithms, Benchmarks, and Evaluation Measures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. ACM.
[99] 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. *Proceedings of the VLDB Endowment* 15, 11 (2022), 2774–2787.
[100] John Paparrizos, Ikraduya Edian, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2022. Fast Adaptive Similarity Search through Variance-Aware Quantization. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2969–2983.
[101] John Paparrizos and Michael J Franklin. 2019. Grail: efficient time-series representation learning. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1762–1777.
[102] John Paparrizos and Luis Gravano. 2015. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 1855–1870.
[103] John Paparrizos and Luis Gravano. 2017. Fast and accurate time-series clustering. *ACM Transactions on Database Systems (TODS)* 42, 2 (2017), 1–49.
[104] 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.
[105] John Paparrizos, Haojun Li, Fan Yang, Kaize Wu, Jens E d'Hondt, and Odysseas Papapetrou. 2024. A survey on time-series distance measures. *arXiv preprint arXiv:2412.20574* (2024).
[106] John Paparrizos, Chunwei Liu, Bruno Barbarioli, Johnny Hwang, Ikraduya Edian, Aaron J Elmore, Michael J Franklin, and Sanjay Krishnan. 2021. VergeDB: A Database for IoT Analytics on Edge Devices.. In *CIDR*.
[107] John Paparrizos, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2020. Debunking four long-standing misconceptions of time-series distance measures. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1887–1905.
[108] John Paparrizos, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2023. Querying Time-Series Data: A Comprehensive Comparison of Distance Measures. *Data Engineering* (2023), 69.
[109] John Paparrizos and Sai Prasanna Teja Reddy. 2023. Odyssey: An Engine Enabling the Time-Series Clustering Journey. *Proceedings of the VLDB Endowment* 16, 12 (2023), 4066–4069.
[110] John Paparrizos and Sai Prasanna Teja Reddy. 2025. Time-Series Clustering: A Comprehensive Study of Data Mining, Machine Learning, and Deep Learning Methods. *Proceedings of the VLDB Endowment* 18, 11 (2025), 4380–4395.
[111] John Paparrizos, Ryen W White, and Eric Horvitz. 2016. Detecting devastating diseases in search logs. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 559–568.
[112] John Paparrizos, Ryen W White, and Eric Horvitz. 2016. Screening for pancreatic adenocarcinoma using signals from web search logs: Feasibility study and results. *Journal of Oncology Practice* 12, 8 (2016), 737–744.
[113] John Paparrizos, Kaize Wu, Aaron Elmore, Christos Faloutsos, and Michael J Franklin. 2023. Accelerating Similarity Search for Elastic Measures: A Study and New Generalization of Lower Bounding Distances. *Proceedings of the VLDB Endowment* 16, 8 (2023), 2019–2032.
[114] John Paparrizos, Fan Yang, and Haojun Li. 2024. Bridging the Gap: A Decade Review of Time-Series Clustering Methods. *arXiv preprint arXiv:2412.20582* (2024).
[115] 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).
[116] 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.
[117] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 427–438.
[118] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Vincent Hassen, Anderson Schneider, et al. 2023. Lag-llama: Towards foundation models for time series forecasting. *arXiv preprint arXiv:2310.08278* (2023).
[119] Shebuti Rayana and Leman Akoglu. 2016. Less is more: Building selective anomaly ensembles. *Acm transactions on knowledge discovery from data (tkdd)* 10, 4 (2016), 1–33.
[120] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 3009–3017.
[121] Ilia Revin, Vadim A Potemkin, Nikita R Balabanov, and Nikolay O Nikitin. 2023. Automated machine learning approach for time series classification pipelines using evolutionary optimization. *Knowledge-based systems* 268 (2023), 110483.
[122] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh international conference on networked sensing systems (INSS)*. IEEE, 233–240.
[123] 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.
[124] Peter J Rousseeuw and Katrien Van Driessen. 1999. A fast algorithm for the minimum covariance determinant estimator. *Technometrics* 41, 3 (1999), 212–223.
[125] 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.
[126] Sebastian Schmidl, Felix Naumann, and Thorsten Papenbrock. 2024. AutoTSAD: Unsupervised Holistic Anomaly Detection for Time Series Data. *Proceedings of the VLDB Endowment* 17, 11 (2024), 2987–3002.
[127] 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.
[128] 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.
[129] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. 2015. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* 104, 1 (2015), 148–175.
[130] Iman Sharafaldin, Arash Habibi Lashkari, Ali A Ghorbani, et al. 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* 1 (2018), 108–116.
[131] 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.
[132] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. 2017. Anomaly detection in streams with extreme value theory. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1067–1075.
[133] 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
[134] 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.
[135] Mingtian Tan, Mike Merrill, Vinayak Gupta, Tim Althoff, and Tom Hartvigsen. 2025. Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems* 37 (2025), 60162–60191.
[136] TheDatumOrg. 2024. TSB-AutoAD. https://github.com/TheDatumOrg/TSB-AutoAD.
[137] 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.
[138] Luan Tran, Liyue Fan, and Cyrus Shahabi. 2016. Distance-based outlier detection in data streams. *Proceedings of the VLDB Endowment* 9, 12 (2016), 1089–1100.
[139] Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. 2022. TranAD: deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment* 15, 6 (2022), 1201–1214.
[140] Joaquin Vanschoren. 2018. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548* (2018).
[141] Bruno Veloso, João Gama, and Benedita Malheiro. 2018. Self hyper-parameter tuning for data streams. In *Discovery Science: 21st International Conference, DS 2018, Limassol, Cyprus, October 29–31, 2018, Proceedings 21*. Springer, 241–255.
[142] Ricardo Vilalta and Youssef Drissi. 2002. A perspective view and survey of meta-learning. *Artificial intelligence review* 18 (2002), 77–95.
[143] 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*.
[144] Renjie Wu and Eamonn J Keogh. 2021. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE transactions on knowledge and data engineering* 35, 3 (2021), 2421–2429.
[145] 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.
[146] 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.

[147] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2021. Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. In *International Conference on Learning Representations*.

[148] 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.

[149] Zhijian Xu, Ailing Zeng, and Qiang Xu. 2023. FITS: Modeling Time Series with $10k$ Parameters. In *The Twelfth International Conference on Learning Representations*.

[150] Takehisa Yairi, Yoshikiyo Kato, and Koichi Hori. 2001. Fault detection by mining association rules from house-keeping data. In *proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Vol. 18. Citeseer, 21.

[151] Fan Yang and John Paparrizos. 2025. SPARTAN: Data-Adaptive Symbolic Time-Series Approximation. In *Proceedings of the ACM on Management of Data*, Vol. 3. Article 220. https://doi.org/10.1145/3725357

[152] 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. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2593–2606.

[153] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*. Ieee, 1317–1322.

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

[155] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems* 36 (2023), 43322–43355.