# SAIL: A Voyage to Symbolic Approximation Solutions for Time-Series Analysis

Fan Yang
The Ohio State University
Columbus, Ohio, USA
yang.7007@osu.edu

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

## ABSTRACT

*Symbolic Approximation*, a dimensionality reduction technique that transforms time series into discrete symbols, has gained increasing attention in various downstream applications. Despite decades of development, there is a noticeable absence of a comprehensive study in this domain, highlighting a need for more in-depth investigation and well-designed exploration tools. To address this gap, we propose **SAIL**, a modular web engine serving two purposes: (i) to provide the first comprehensive study on 7 state-of-the-art methods over 100+ time-series datasets, the largest study in this area; (ii) to evaluate the performance of a recently proposed solution, SPARTAN, that solves two core problems. First, SPARTAN exploits intrinsic dimensionality reduction to effectively model the underlying data distribution for approximation. Second, SPARTAN dynamically allocates alphabet sizes per segment, recognizing the non-uniform distribution of information in practice. Through its interactive interface, SAIL enables users to visualize and explore quantitative assessments across various methods, datasets, and analytical tasks. SAIL's exploration reveals that (i) while SAX variants outperform SAX by sacrificing storage, none surpass SAX under the same budget, reinforcing it as a strong baseline; SFA is the only existing method that consistently outperforms SAX within the same budget; and (ii) across diverse scenarios, SPARTAN outperforms competing methods in all evaluated tasks significantly, including classification, clustering, indexing, and anomaly detection, without incurring additional storage or runtime overhead. Overall, SAIL not only facilitates the most comprehensive studies in this field but also provides new insights and concrete solutions for future research. We release the SAIL web engine at https://saildemo.streamlit.app/.

**PVLDB Reference Format:**
Fan Yang and John Paparrizos. SAIL: A Voyage to Symbolic Approximation Solutions for Time-Series Analysis. PVLDB, 18(12): 5419 - 5422, 2025.
doi:10.14778/3750601.3750686

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

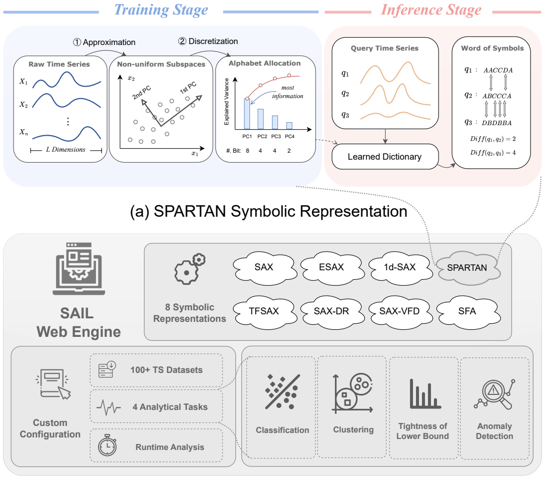
**Table 1: Features of symbolic approximation methods. Complexity results are estimated by $n$ time series with length $m$ and word length $\omega$ for final representation. $s$ denotes the number of downsampled data for training.**

| Method | Lower Bounding | Data-dependent Approximation | Dynamic Discretization | Complexity (training-stage) | Complexity (inference-stage) |
|---|---|---|---|---|---|
| SAX [7] | ✓ | – | – | $O(nm)$ | $O(nm)$ |
| ESAX [10] | – | – | – | $O(nm)$ | $O(nm)$ |
| TFSAX [23] | – | – | – | $O(nm)$ | $O(nm)$ |
| SAX-DR [6] | – | – | – | $O(nm)$ | $O(nm)$ |
| SAX-VFD [21] | – | ✓ | – | $O(nm\log(nm))$ | $O(nm\log(nm))$ |
| 1d-SAX [11] | – | – | – | $O(nm)$ | $O(nm)$ |
| SFA [18] | ✓ | – | – | $O(nm\log(m))$ | $O(nm\log(m))$ |
| ***Newly Proposed Symbolic Representation Solution*** | | | | | |
| **SPARTAN [22]** | ✓ | ✓ | ✓ | $O(nm^2)$ | $O(nm\omega)$ |
| **SPARTAN-R [22]** | ✓ | ✓ | ✓ | $O(nm\omega)$ | $O(nm\omega)$ |
| **SPARTAN-S [22]** | ✓ | ✓ | ✓ | $O(sm\omega)$ | $O(nm\omega)$ |

## 1 INTRODUCTION

With the expansion of the Internet of Things (IoT) sensor devices, web data collection, and remote sensing, time-series data mining techniques have been widely applied in diverse downstream tasks [4, 9, 13, 15, 16]. For its simplicity and scalability, *symbolic approximation*, a dimensionality reduction technique for efficient analysis, has received increasing attention [12, 20]. A symbolic representation is produced to transform a time series into a string of discrete symbols, which offers benefits from both the dimensionality reduction process and high interpretability. Over the past two decades, symbolic methods have become a subroutine in diverse analytical tasks, such as dictionary classifiers [8, 17] and indexing [2]. Despite decades of progress, a non-negligible gap remains in this field regarding the comprehensive study of time-series symbolic solutions [7, 10, 11, 18, 21, 23]. We observed that most present symbolic approximation methods (hereafter referred to as symbolic methods) fail to effectively model the underlying distribution and the disproportionate importance of each segment. In particular, current strategies share a key naivete: they assume only a *single* alphabet of a *single* size for each segment, ignoring the non-uniform nature, e.g., most energy of the signal may concentrate on a few segments. In practice, this uniform balance strategy in prior works may potentially waste valuable encoding budget on less important information, leading to a degradation in their representation efficiency when adapted to various domains. Compared with data-agnostic approaches focusing on a single time series each time, our recent work SPARTAN [22] proposes a data-adaptive solution to construct uncorrelated latent dimensions for approximation.
Moreover, SPARTAN enables non-uniform budget allocation by considering the importance of each dimension under the same budget, achieved through a novel dynamic programming approach. Table 1 summarizes the crucial features of all surveyed methods.

Proceedings of the VLDB Endowment, Vol. 18, No. 12 ISSN 2150-8097.
doi:10.14778/3750601.3750686



(a) SPARTAN Symbolic Representation

(b) SAIL Web Engine

**Figure 1: An overview of SPARTAN symbolic representation and the proposed SAIL web engine.**

In this work, we introduce **SAIL**, a modular web engine for time-series **S**ymbolic representation **A**nalysis and **I**nteractive exp**L**oration. SAIL provides the first comprehensive evaluation studies on 7 current state-of-the-art methods over 100+ time series datasets, the largest study in this field. As illustrated in Figure 1, SAIL facilitates interactive exploration through flexible choices of methods and datasets. Four critical analytical tasks are supported for assessing the representation quality, including classification, clustering, anomaly detection, and indexing (proxy by the tightness of lower bound (TLB) [7]). Through interactive exploration, SAIL reveals that (i) while SAX variants improve performance by increasing storage, none surpass SAX under the same budget, reinforcing it as a strong baseline, and SFA remains the only baseline method that consistently outperforms SAX under this constraint; and (ii) SPARTAN demonstrates superior representation power over top methods across all evaluated tasks by leveraging intrinsic dimensionality reduction properties and adapting to the varying importance of different dimensions. In general, SAIL takes a significant step further in unveiling the current landscape of time-series symbolic solutions and provides valuable insights for future studies.

## 2 PRELIMINARIES

In this section, we first present the current state-of-the-art methods, and then introduce SPARTAN, a recently proposed data-adaptive solution integrated as the core component of SAIL web engine.

### 2.1 Symbolic Representation

Symbolic representation serves as an efficient tool to construct a lightweight representation for time series in real-time [2]. Typically, symbolic representation methods transform raw time series into a sequence of discrete symbols (e.g., "ACBACA"), where specific symbol combinations form "words" analogous to those in natural language–capturing representative shapes or recurring patterns in the data. Owing to their discrete and symbolic nature, symbolic methods provide high interpretability through human-readable symbols, leverage a wide range of text-based techniques to facilitate the discovery of meaningful patterns, and offer a lower-bounding property that prevents false dismissals in similarity search tasks [7, 18, 22]. In recent years, symbolic methods have become a core component in a variety of analytical tasks, such as dictionary-based classification, anomaly detection, and indexing [17, 19, 20].

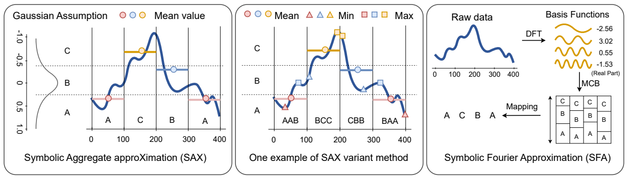


**Figure 2: Examples of SAX, SAX variant method, and SFA.**

### 2.2 Symbolic Representation Methods

As one of the most fundamental solutions in this domain, SAX [7] has achieved great success in diverse tasks for its simplicity and efficiency. Given a time series, SAX reduces the dimensionality by Piecewise Aggregate Approximation (PAA) and then maps the mean value of each segment to a discrete symbol according to a predefined look-up table. As shown in Figure 2, diverse SAX variants have been proposed with additional information such as shape and statistical features [10, 11, 21, 23]. In contrast, SFA [18], leverages Discrete Fourier Transform (DFT) to capture the frequency information from the Fourier domain. In addition, Multiple Coefficient Binning (MCB) is also proposed as the standard discretization technique for alphabet dictionaries. Table 1 summarizes the crucial features of all surveyed methods. In the following, we are going to review the recently proposed SPARTAN symbolic representation [22].

### 2.3 SPARTAN: Algorithm Overview

SPARTAN [22], a recently proposed symbolic method, is integrated as a core component of the SAIL web engine. In the following, we review the two main stages for training: approximation and discretization. During inference, the time series can be transformed on-the-fly using a pretrained alphabet dictionary (Figure 1 (a)).

**Approximation:** Existing methods often fall short in the approximation process due to two key limitations: (i) approximation techniques like PAA and DFT do not leverage information from the entire dataset, limiting adaptability across domains; and (ii) they fail to account for the non-uniform importance of segments in practical settings. To address these issues, SPARTAN leverages intrinsic dimensionality reduction to identify and prioritize segments based on their importance. Specifically, it ranks segments by descending variance, enabling more informative symbol generation during discretization. Observations on the widely used UCR dataset [3] reveal that most information is concentrated in a few top principal components, highlighting the need for adaptive methods like SPARTAN. To reduce training time, SPARTAN supports accelerated variants using randomized solvers [5] and sampling strategies, effectively pruning computation costs–especially on large scale data.

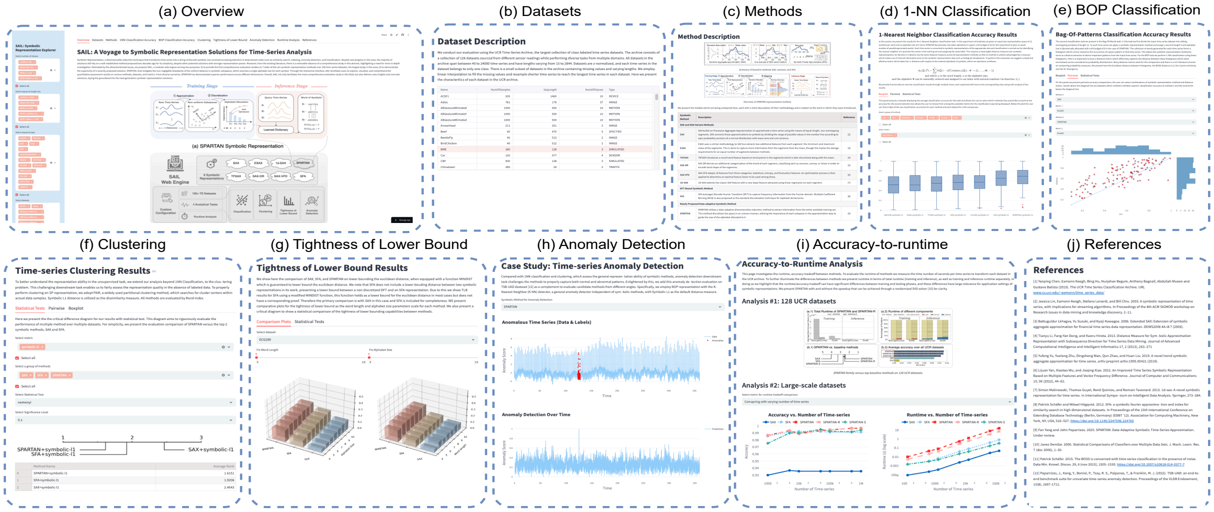

(a) Overview
(b) Datasets
(c) Methods
(d) 1-NN Classification
(e) BOP Classification
(f) Clustering
(g) Tightness of Lower Bound
(h) Anomaly Detection
(i) Accuracy-to-runtime
(j) References
SAIL: A Voyage to Symbolic Representation Solutions for Time-Series Analysis
Dataset Description
Method Description
1-Nearest Neighbor Classification Accuracy Results
Bag-Of-Patterns Classification Accuracy Results
Time-series Clustering Results
Tightness of Lower Bound Results
Case Study: Time-series Anomaly Detection
Accuracy-to-Runtime Analysis
Analysis #1: 128 UCR datasets
Analysis #2: Large-scale datasets
References
SPARTAN+symbolic-l1
SFA+symbolic-l1
SAX+symbolic-l1

Figure 3: The examples of interactive interfaces in SAIL web engine.

**Discretization:** Previous solutions typically allocate an equal-sized alphabet to each subspace, often wasting capacity on less informative dimensions. To address this, SPARTAN introduces *Dynamic Alphabet Allocation*, which assigns bits proportionally to the importance across dimensions. However, trivial solutions may simply allocate all bits to the first few dimensions. To avoid this, SPARTAN employs a dynamic programming solution to efficiently solve a constrained optimization problem for allocating the budget across dimensions. With a pre-trained alphabet dictionary, time series can then be symbolized on the fly for real-time inference.

## 3 SAIL: SYSTEM OVERVIEW

To help users delve into this field, SAIL web engine[1] provides interactive interfaces and diverse scenarios, supported by Python 3.9 and the Streamlit [1]. As shown in Figure 3, the GUI of SAIL incorporates 10 major frames for different tasks.

**Classification:** This frame facilitates the evaluation on the widely-used downstream task, 1NN Classification, over 128 UCR datasets. A generic distance measure, i.e., Symbolic-$L_1$ distance, is adopted. Users have flexible options for selecting datasets and evaluated methods. To help users comprehend the results, a box plot is provided to visualize the classification accuracy performance over selected datasets, followed by the pairwise comparison and critical diagram (CD) with global rankings (we utilize Friedman test followed by the post-hoc Nemenyi test with 95% confidence level). In addition, SAIL also supports Bag-of-Patterns (BOP) [17, 22] as the primary representation format for each time series. BOP collects a set of symbol words from sliding windows across raw dimensions and count their distribution, which focuses more on the local patterns. We utilize Euclidean distance (ED) for measuring the dissimilarity between BOPs and perform 1NN classification.

**Clustering:** This frame assesses the unsupervised representation quality of each method. Similar to the classification frames, we adopt the UCR dataset [3] and utilize the same class labels for evaluation. We utilize partition around medoids (PAM) method for all methods, and measure the distance by Symbolic-$L_1$. Boxplots, pairwise comparison and critical diagrams have been provided.

**Tightness of Lower Bound:** To quantify the pruning power, SAIL adopts the tightness of lower bound (TLB) [7] as the evaluation metric, which quantifies the ratio between the dissimilarity in the symbolic space and the Euclidean distance in the original space [7]. A larger TLB (TLB $< 1$) shows a smaller gap between the symbolic and true distance, demonstrating greater pruning power. This lower bounding guarantee, preventing false dismissals when searching the data, has also become a desirable property in indexing and similarity search [2, 18] (also be seen as a proxy for indexing capability). Through the interactive interface, users can visualize the trend of TLB with increasing/decreasing parameters, e.g., word length $\omega$ and alphabet size $\alpha$, and validate the findings with statistical tests.

**Anomaly Detection:** Anomaly detection downstream task evaluates the ability of methods to accurately capture both normal and abnormal patterns. Motivated by this, we include an anomaly detection case study with symbolic representation + K-NN on the TSB-UAD dataset [14] to provide a complementary assessment from multiple perspectives. SAIL offers an interactive GUI for visualizing raw data with labels alongside the predicted anomaly scores, enabling users to compare SPARTAN with top baseline methods. The interface also supports zooming in and out for detailed exploration.

**Runtime Analysis:** SAIL facilitates users in performing accuracy-to-runtime analysis, comparing SPARTAN with current state of the arts. The theoretical time complexity of each method can be found in Table 1. First, SAIL conducts a 1-NN classification accuracy analysis, comparing the SPARTAN family (including its accelerated versions) with top baseline methods across 128 UCR datasets, demonstrating its robustness. Next, SAIL allows users to explore the scalability of each evaluated method on large-scale datasets (synthetic CBF) to simulate practical scenarios. Users can customize comparisons based on the number of time series or time-series lengths and visualize the results through line charts.

[1] Available online: https://saildemo.streamlit.app/

# 4 DEMONSTRATION OVERVIEW

This demo showcases three core scenarios of the SAIL system, which facilitate user exploration across different dimensions of each symbolic methods: (i) to benchmark the discriminative power of each symbolic representation ($\mathcal{S}1$); (ii) to explore the pruning power ($\mathcal{S}2$); (iii) to investigate the trade-off between accuracy and runtime ($\mathcal{S}3$). Users can visualize and delve into the comprehensive evaluation study on 7 state-of-the-art solutions plus SPARTAN, which unveils the landscape of this domain. Notably, all methods are evaluated *under the same budget* to ensure fairness [22].

**$\mathcal{S}1$: Benchmark the discriminative power.** We present an explorative study in three discriminative tasks: classification, clustering, and anomaly detection (Figure 3). Notably, SAIL reveals that while SAX variants often report superior performance, they typically incur higher storage costs. *Under equivalent storage budgets, none significantly outperform SAX, reinforcing its status as a strong baseline. Among existing methods, only SFA consistently achieves superior performance under the same budget.* All conclusions are validated by statistical testing. Leveraging data-adaptive approximation and Dynamic Alphabet Allocation, recently proposed SPARTAN consistently outperforms the top methods in both 1NN and BOP classification tasks, offering a more effective representation for capturing discriminative patterns. This superiority is further supported by clustering and anomaly detection. Through interactive case studies in anomaly detection, SAIL helps identify the vulnerabilities in existing solutions, including failure cases likely caused by their reliance on PAA quality and sensitivity to frequency variations. In contrast, SPARTAN prioritizes the informative dimensions with dynamic budget allocation, resulting in a more effective representation that accurately captures both normal and abnormal patterns.

**$\mathcal{S}2$: Assess the pruning power.** In the preliminary exploration by SAIL, we observe that most methods exhibit violations (TLB > 1), as shown in Table 1 – a critical issue overlooked in prior studies. Notably, only SPARTAN, SAX, and SFA consistently guarantee the TLB property, as demonstrated in the experiments. Within the `Pruning Power Assessment` frame, users can easily visualize the trend of TLB across different alphabet sizes $\alpha$ and word lengths $\omega$, from 3D bar plots of TLB values of different datasets. Both comparison plots and statistical tests by SAIL reveal that SPARTAN consistently outperforms existing top approaches, demonstrating superior pruning power in the symbolic space.

**$\mathcal{S}3$: Investigate the trade-off between accuracy and runtime.** Figure 3 depicts that, in the `Runtime Analysis` frame, SAIL first demonstrates the robustness of (1-NN classification) accuracy performance of SPARTAN versus the top two methods, SAX and SFA, over 128 datasets, showing with both average accuracy and statistical test. Both accelerated versions, SPARTAN-R (randomized strategy) and SPARTAN-S (sampling strategy), have demonstrated the leading performance compared to the baseline methods while significantly reducing computation costs compared to the original version. In the second analysis on large-scale datasets, users can visually compare the accuracy and runtime for each method with increasing data dimensions, e.g., 1M time series samples. Notably, SPARTAN's accelerated variant provides up to a 2x speedup over SFA on large-scale databases containing millions of time series, while preserving representation quality in the symbolic space.

# 5 CONCLUSION

In this paper, we introduce SAIL, a modular web engine that allows users to explore the time-series symbolic representation field. With an interactive interface, users are enabled to explore and comprehend a recently proposed method SPARTAN along with 7 strong baselines on 4 analytical tasks. Across diverse settings, SPARTAN has shown superior performance across all dimensions, without introducing storage or computation overhead. We hope this demonstration system offers users a valuable opportunity to delve into this field and paves the way for future research.

# REFERENCES

[1] [n.d.]. *Streamlit documentation.* https://docs.streamlit.io

[2] Alessandro Camerra, Themis Palpanas, Jin Shieh, and Eamonn Keogh. 2010. iSAX 2.0: Indexing and mining one billion time series. In *Proc. IEEE ICDM*. 58–67.

[3] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica* 6, 6 (2019), 1293–1305.

[4] Jens E d'Hondt, Haojun Li, Fan Yang, Odysseas Papapetrou, and John Paparrizos. 2025. A Structured Study of Multivariate Time-Series Distance Measures. *SIGMOD'25* 3, 3 (2025), 1–29.

[5] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review* 53, 2 (2011), 217–288.

[6] Tianyu Li, Fang-Yan Dong, and Kaoru Hirota. 2013. Distance Measure for Symbolic Approximation Representation with Subsequence Direction for Time Series Data Mining. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 17, 2 (2013), 263–271.

[7] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. 2003. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In *Proc. DMKD*. 2–11.

[8] Jessica Lin, Rohan Khade, and Yuan Li. 2012. Rotation-invariant similarity in time series using bag-of-patterns representation. *Journal of Intelligent Information Systems* 39 (2012), 287 – 315. https://api.semanticscholar.org/CorpusID:873260

[9] Qinghua Liu and John Paparrizos. 2024. The Elephant in the Room: Towards A Reliable Time-Series Anomaly Detection Benchmark. In *NeurIPS 2024*.

[10] Battuguldur Lkhagva, Yu Suzuki, and Kyoji Kawagoe. 2006. Extended SAX: Extension of symbolic aggregate approximation for financial time series data representation. *DEWS2006 4A-i8* 7 (2006).

[11] Simon Malinowski, Thomas Guyet, René Quiniou, and Romain Tavenard. 2013. 1d-sax: A novel symbolic representation for time series. In *International Symposium on Intelligent Data Analysis*. Springer, 273–284.

[12] Thach Le Nguyen and Georgiana Ifrim. 2022. MrSQM: Fast Time Series Classification with Symbolic Representations. arXiv:2109.01036 [cs.LG]

[13] John Paparrizos and Luis Gravano. 2015. k-shape: Efficient and accurate clustering of time series. In *SIGMOD'15*. 1855–1870.

[14] 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. *VLDB* 15, 8 (2022), 1697–1711.

[15] 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).

[16] 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).

[17] Patrick Schäfer. 2015. The BOSS is concerned with time series classification in the presence of noise. *DMKD* 29, 6 (2015), 1505–1530.

[18] Patrick Schäfer and Mikael Högqvist. 2012. SFA: a symbolic fourier approximation and index for similarity search in high dimensional datasets. In *EDBT12*. 516–527.

[19] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P Boedihardjo, Crystal Chen, and Susan Frankenstein. 2015. Time series anomaly discovery with grammar-based compression.. In *EDBT*. 481–492.

[20] Jin Shieh and Eamonn Keogh. 2008. iSAX: Indexing and mining terabyte-sized time series. In *Proc. KDD*. 623–631.

[21] Lijuan Yan, Xiaotao Wu, and Jiaqing Xiao. 2022. An Improved Time Series Symbolic Representation Based on Multiple Features and Vector Frequency Difference. *Journal of Computer and Communications* 10, 06 (2022), 44–62.

[22] Fan Yang and John Paparrizos. 2025. SPARTAN: Data-Adaptive Symbolic Time-Series Approximation. *SIGMOD'25* 3, 3 (2025), 1–30.

[23] Yufeng Yu, Yuelong Zhu, Dingsheng Wan, Qun Zhao, and Huan Liu. 2019. A novel trend symbolic aggregate approximation for time series. *arXiv preprint arXiv:1905.00421* (2019).