Homework: Research article replication - NoSQL

Prepared by

Anna Krysta

Master M2 IASD – Université PSL (Paris Sciences & Lettres)

March 15, 2025

1 Introduction

In recent years, the surge in large-scale time series data across fields like science, engineering, and finance has created a pressing demand for more efficient ways to index and retrieve this information. While traditional methods like R-trees and KD-trees[3] have been useful, they struggle to keep up as data volumes grow. The iSAX[7] framework addresses these limitations by introducing a multi-resolution symbolic representation that enables hierarchical indexing and rapid similarity search. In this report, we'll explore how effective iSAX really is, replicate its experiments, and take a closer look at its impact on the field of data mining.

2 Methodology and results

The iSAX method extends the classic SAX[5] representation by enabling multi-resolution indexing, allowing for both approximate and exact search in massive datasets. Unlike SAX, which assigns a fixed-length symbolic representation to time series, iSAX allows variable-length symbolic words that improve hierarchical partitioning and retrieval efficiency.

The core aspects of iSAX include:

- **Hierarchical indexing:** The method constructs a tree-based index where each node represents time series segments at varying levels of granularity.
- **Lower bounding guarantees:** The representation ensures that distance calculations at a lower resolution do not underestimate true distances, thereby preventing false dismissals.
- **Efficient disk-based storage:** The method structures indexed data to minimize redundant access, significantly reducing retrieval latency.

The experimental evaluation in the original paper demonstrates the efficiency of iSAX across several large datasets. The results indicate that approximate search can be performed in constant time with a single disk access, while exact search remains feasible within reasonable computational bounds. The authors validate these claims through benchmark comparisons against traditional indexing methods.

3 Method

The iSAX method builds on the classic SAX representation but extends it to support variable granularity and extensible hashing. The key steps in iSAX are:

1. **Normalization:** Time series data is normalized to have a mean of zero and a standard deviation of one.
2. **PAA (Piecewise Aggregate Approximation):** The time series is divided into w segments, and the mean value of each segment is calculated.
3. **Discretization:** The PAA values are discretized into symbols using breakpoints derived from a Gaussian distribution.
4. **Multi-Resolution Representation:** iSAX allows for variable cardinality (number of symbols) in the representation, enabling hierarchical indexing.

3.1 Mathematical basics of iSAX

The PAA representation of a time series is given by:

$$\bar{t}_i = \frac{1}{w} \sum_{j \in \text{segment } i} t_j$$

where $\bar{t}_i$ is the mean value of the i-th segment. The PAA values are discretized into symbols using breakpoints β derived from a Gaussian distribution $N(0,1)$. The breakpoints are chosen such that the area under the Gaussian curve between β_i and β_{i+1} is $1/a$, where a is the cardinality.

3.2 Lower bounding distance of iSAX

Lower bounding is a critical component of the iSAX representation because it enables efficient similarity search by pruning the search space without introducing false dismissals. Specifically, the lower bound ensures that any time series discarded during the search process cannot be closer to the query than the current best match, guaranteeing that no true matches are missed. This is particularly important for large-scale datasets, where exhaustive search is computationally infeasible.

The distance between two iSAX words $\mathbf{T}^a$ and $\mathbf{S}^b$, where a and b are the cardinalities, is given by:

$$MINDIST(\mathbf{T}^a, \mathbf{S}^b) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} (dist(t_i, s'_i))^2}$$

where $dist(t_i, s'_i)$ is the distance between the symbols t_i and s'_i in the iSAX word. If $b < a$, the symbols in $\mathbf{S}^b$ are promoted to the cardinality a by filling in the missing bits in a way that minimizes the distance, ensuring that the resulting distance is still a valid lower bound. This promotion process preserves the hierarchical structure of iSAX, allowing comparisons between representations of different resolutions.

The lower bound $MINDIST$ bounds the true Euclidean distance between the original time series T and S. Specifically, it satisfies:

$$MINDIST(\mathbf{T}^a, \mathbf{S}^b) \leq D(T, S),$$

where $D(T, S)$ is the true Euclidean distance. This property ensures that the search process is both efficient and accurate, as it allows iSAX to prune large portions of the dataset that cannot contain better matches than the current best candidate. The tightness of the lower bound improves as the cardinality increases, approaching the true Euclidean distance, which is crucial for maintaining the accuracy of the search while minimizing computational overhead. This combination of efficiency and accuracy makes iSAX highly scalable for large datasets, enabling fast exact and approximate search.

4 Strong points

The paper introduces iSAX, a novel multi-resolution symbolic representation that significantly advances the field of time series indexing and mining by enabling the handling of datasets several orders of magnitude larger than previously considered. Traditional approaches have struggled with datasets beyond the megabyte level, but iSAX successfully indexes and mines terabyte-sized datasets, which is a monumental leap in scalability. This is achieved through a modification of the SAX representation, allowing for extensible hashing and multi-resolution capabilities. The multi-resolution property of iSAX ensures that time series can be indexed with zero overlap at leaf nodes, a feature not supported by other spatial access methods like R-trees. This scalability is crucial for modern applications in science, engineering, and business, where massive datasets are increasingly common.

Another strong point of the paper is its ability to perform both fast exact search and ultra-fast approximate search, which is essential for handling the vast amounts of data efficiently. The iSAX representation allows for the use of data structures and algorithms that are not typically applicable to real-valued data, such as suffix trees, hashing, and Markov models. This symbolic representation not only enhances the speed of search operations but also ensures that the approach is both practical and easy to adopt, as it does not require specialized databases or file managers. The paper demonstrates that iSAX can index up to 100 million time series, a feat that is at least two orders of magnitude larger than any dataset previously considered in the literature. This capability is further validated through extensive experiments, showing that iSAX can retrieve highly accurate results with minimal disk access, making it a powerful tool for real-world applications.

The next strength of the iSAX method is its ability to eliminate overlap at leaf nodes, ensuring efficient storage and retrieval of time series data. Traditional indexing structures, such as R-trees or other spatial access methods, often suffer from node overlap, which leads to redundant searches and inefficient disk access. In contrast, iSAX utilizes a multi-resolution symbolic representation that enables hierarchical partitioning of the dataset, ensuring that each time series is uniquely assigned to a distinct leaf node without duplication. This property not only optimizes storage by preventing redundant data entries but also enhances retrieval efficiency by reducing the number of disk accesses required to locate relevant time series. As a result, query performance is significantly improved, making iSAX well-suited for handling massive time series databases in a scalable manner.

5 Weak points

One potential drawback of iSAX is its sensitivity to parameter selection. The performance of iSAX heavily depends on parameters such as word length, base cardinality, and threshold values. Selecting optimal parameters often requires domain expertise or extensive trial and error, which can be a barrier for users without specialized knowledge. If these parameters are not chosen carefully, the indexing process may become inefficient, or the approximate search results may lack accuracy, undermining the system's overall effectiveness.

Another limitation is that iSAX may not perform as well with highly variable or noisy datasets. While it excels with structured time series, its symbolic representation can lose fine-grained details, particularly when dealing with non-stationary or irregular data. The fixed breakpoints used in SAX can introduce distortions when applied to datasets with erratic patterns, potentially leading to errors in similarity searches and reducing the reliability of the results.

Lastly, the approximate search mechanism in iSAX, while ultra-fast, can sometimes produce suboptimal initial results. This necessitates further refinement through exact search, which can offset some of the performance benefits. If the approximate search fails to identify a close match early in the process, the overall efficiency gains compared to a brute-force approach may diminish. The success of the approximate search largely hinges on how well the indexed structure aligns with the underlying data distribution, which may not always be guaranteed.

6 Research problem addressed by iSAX

iSAX was developed to tackle a critical problem in data management: the inability of traditional indexing structures to efficiently handle massive time series datasets. As data collection rates continue to accelerate in domains such as finance, bioinformatics, and sensor networks, existing indexing methods fail to scale accordingly. iSAX provides a solution by introducing a hierarchical, multi-resolution symbolic representation that maintains computational feasibility even for terabyte-sized datasets.

7 Comparison with state-of-the-art methods

Before the introduction of iSAX, several methods were developed to tackle the challenges of time series indexing, but each had significant limitations when applied to large-scale datasets:

- **R-trees and KD-trees[3]:** These traditional spatial indexing methods are effective for low-dimensional data but struggle with the high dimensionality and complexity of time series. Issues such as node overlap and inefficient disk access become pronounced as dataset sizes grow, leading to slower query performance and increased storage overhead. As time series length increases, these methods often degrade in efficiency, forcing reliance on brute-force searches, which are impractical for large-scale analysis.
- **Piecewise Aggregate Approximation (PAA)[4]:** While PAA reduces dimensionality effectively, it lacks a hierarchical indexing structure, which limits its ability to scale to massive datasets. Without a multi-level organization, PAA struggles to maintain efficiency as data volume grows.
- **Discrete Wavelet Transform (DWT)[1] and Discrete Fourier Transform (DFT)[3]:** Both DWT and DFT offer strong compression capabilities, but they come with significant computational overhead during indexing and retrieval. This overhead can hinder their performance, especially when dealing with large datasets where efficiency is critical.

iSAX addresses these limitations by introducing a symbolic, multi-resolution approach that combines the strengths of dimensionality reduction with a hierarchical indexing structure. Unlike traditional methods, iSAX eliminates node overlap and efficiently balances approximate and exact search, enabling scalable and accurate indexing even for terabyte-sized datasets. By leveraging progressive refinement, iSAX achieves high performance in both efficiency and accuracy, making it a significant advancement over previous state-of-the-art methods.

8 Future research

Future research on iSAX can explore several key directions to further enhance its utility and efficiency. One major area of improvement is real-time streaming capabilities, where adapting iSAX to dynamically index and query continuously incoming time series data with minimal latency could significantly impact domains like financial markets and IoT sensor networks. Additionally, optimizing AI-driven predictive analytics by integrating iSAX with deep learning models could improve forecasting accuracy and anomaly detection, leveraging its symbolic representation for efficient training and inference. Another crucial avenue is advancing compression techniques while preserving search accuracy, enabling more compact storage of massive datasets without sacrificing retrieval performance. Furthermore, exploring cross-domain applications such as genomics, climate modeling, and industrial monitoring can validate and refine iSAX's adaptability across diverse data types. Finally, refining hybrid indexing approaches that combine symbolic representations with probabilistic or graph-based indexing methods could lead to even more scalable and precise time series retrieval systems.

9 Evaluation of experiments

The authors of the iSAX paper provided a website where all the datasets used in their experiments are available, along with the raw numbers of the results and additional experiments that did not have the chance to be included in the paper.

The paper makes several key claims regarding the iSAX representation and its ability to index and mine massive time series datasets. The primary claims include: (1) iSAX enables indexing of datasets that are orders of magnitude larger than those considered in prior literature, (2) iSAX supports both fast exact search and ultra-fast approximate search, and (3) iSAX can be integrated into data mining algorithms to allow for exact mining of massive datasets containing millions of time series. The experimental results in the paper effectively cover these claims. For instance, the authors demonstrate the scalability of iSAX by indexing datasets with up to 100 million time series, which is significantly larger than previous benchmarks. They also show that approximate search can retrieve highly accurate results with minimal disk access, while exact search, though more computationally intensive, is still feasible for large datasets. The experiments further validate the utility of iSAX in data mining tasks, such as Time Series Set Difference (TSSD) and batch nearest neighbor search, where the combination of approximate and exact search leads to efficient and accurate results. These experiments collectively substantiate the claims by providing empirical evidence of iSAX's scalability, speed, and applicability to real-world data mining problems.

10 Missing related work

I did not manage to find the PDF file of "On The Marriage of Lp-norms and Edit Distance"[2], and I only read the abstract. However, based on the abstract, it is evident that the iSAX paper should have cited this work. The abstract highlights the introduction of ERP (Edit distance with Real Penalty), a novel distance function that combines the strengths of Lp-norms and edit distance, offering a metric that supports local time shifting, a limitation of traditional Lp-norms like Euclidean distance. This is directly relevant to the iSAX paper, which focuses on Euclidean distance for large-scale time series indexing but does not explore alternative distance measures that could handle local time shifting while maintaining metric properties. Additionally, the abstract discusses efficient pruning strategies for ERP, including the use of a lower bound that can be indexed with a B+ tree, which aligns with the iSAX paper's emphasis on scalable indexing and search. The omission of this paper leaves a gap in the iSAX paper's exploration of alternative methods for handling time series data, particularly in scenarios where local time shifting is a concern.

I did not find the entire paper "Tuning Time Series Queries in Finance: Case Studies and Recommendations"[6], and I only had access to the first page. However, based on the abstract and the introductory content, it is clear that the iSAX paper should have cited this work. The paper by Dennis Shasha delves into the unique challenges of working with time series data in financial databases, emphasizing the critical importance of speed, reliability, and subsecond response times in high-stakes financial environments. While the iSAX paper focuses on indexing and mining massive time series datasets, it does not address the specific demands of financial applications, which are a major real-world use case for time series analysis. Shasha's paper highlights the need for efficient time series querying in finance, where even minor delays or inefficiencies can have significant financial consequences.

11 Implementation & experiments recreation

In this section, we discuss the implementation of the proposed method. This study involves the experimental evaluation of the iSAX method for symbolic representation of time series data. However, upon investigation, we found that the original implementation used by the authors was not publicly accessible. The paper did not provide source code or supplementary materials detailing the implementation. Consequently, we developed our own implementation to replicate the experimental results.
The implemented method follows the description provided in the paper, utilizing the following key steps:

- **Normalization:** The input time series is normalized to ensure a zero mean and unit variance, making it robust to amplitude variations.
- **Piecewise Aggregate Approximation (PAA):** The time series is segmented into equal-length sections, and an average value is computed for each segment.
- **iSAX Transformation:** The PAA segments are mapped to symbolic representations using breakpoints derived from the inverse cumulative distribution function (CDF) of a standard normal distribution.
- **Distance Computation:** A distance function, MINDIST, is implemented to compute the lower bound distance between two iSAX words, ensuring accurate similarity measurements.

To validate our implementation, we replicated key figures from the original paper, including Figures 2 & 3 and 4, as well as generating breakpoints for different cardinalities to match Table 2.

11.1 Recreation of Figure 4 - Tightness of Lower Bound across cardinalities (SAX)

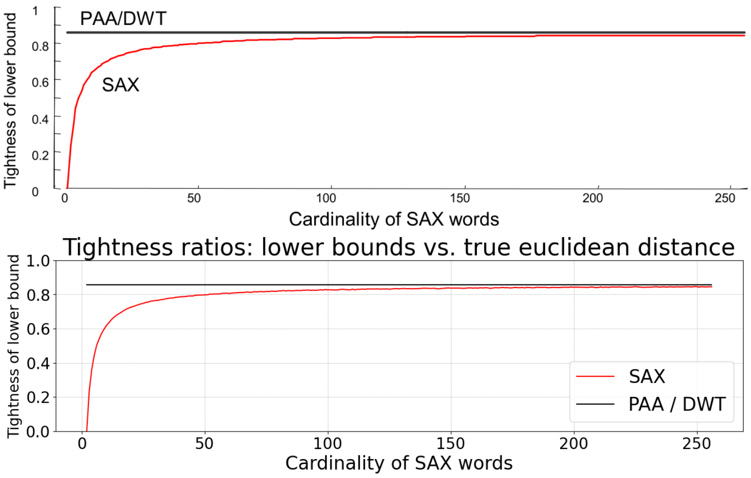

Figure 1: Comparison of Figure 4 from the original paper (top) and our recreated experiment (bottom).

The Figure 4 from the paper in question evaluates the effect of varying SAX cardinalities on the tightness of the lower bound. In this case, we successfully recreated the exact results as presented in the paper, confirming the correctness of our implementation.

For this experiment, we generated random walk time series of length 256 and applied SAX with different cardinalities. We computed the tightness of the lower bound using the MINDIST function and compared it to alternative dimensionality reduction techniques such as DWT and PAA. Both of the methods have the same TLB. The results indicate that as the SAX cardinality increases, the lower bound tightness approaches that of DWT and PAA methods while maintaining the advantages of symbolic representation. Our implementation followed the exact methodology described in the paper, ensuring an accurate reproduction of Figure 4.

Furthermore, we conducted two primary experiments aligned with the paper:

11.2 Recreation of first experiment (Figure 6) - Tightness of Lower Bound

The first experiment evaluates the tightness of the lower bound across different dimensionality reduction techniques, comparing iSAX with other methods such as DCT, APCA, PAA, CHEB, and IPLA. The tightness of the lower bound (TLB) is computed as:

$$TLB = \frac{LowerBoundDist(T,S)}{TrueEuclideanDist(T,S)} \tag{1}$$

where T and S are random time series samples. We reproduced this experiment by generating synthetic datasets and calculating TLB values over various time series representations on the Koski ECG dataset. This dataset was downloaded from the oficial iSAX home page www.cs.ucr.edu/ eamonn/iSAX/iSAX.html. However, we did not manage to exactly replicate the numerical results presented in the original paper. Nevertheless, our results successfully reproduced the same trend: the lower bound for iSAX is consistently greater (closer to 1) than that of any other method we compared it with, confirming the key claim of the paper.
One possible reason for the discrepancy in numerical values could be the method used to calculate *LowerBoundDist*. In our implementation, we used the L2 norm for this calculation, but it is unclear from the paper whether the authors employed the same approach or a different metric. Additionally, the paper does not explicitly state how the calculation of *LowerBoundDist* was performed for the other comparison methods, leaving some ambiguity in how the values were derived.

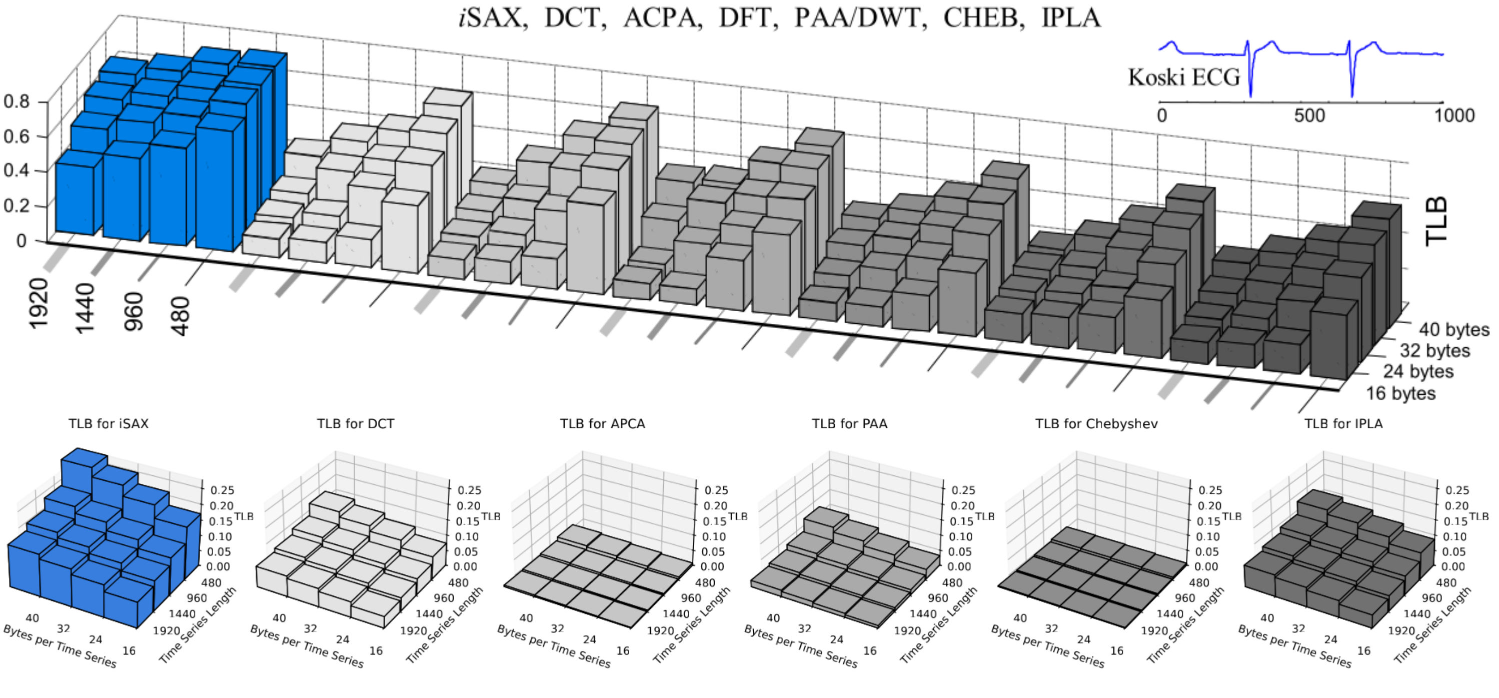

Figure 2: Comparison of Figure 6 from the original paper (top) and our recreated experiment (bottom).

11.3 Recreation of second experiment (Figure 8) - Indexing Massive Datasets

The second experiment (Figure 8) demonstrates iSAX's ability to efficiently index massive time series datasets. The study evaluates how well iSAX can reduce the number of disk accesses required for nearest-neighbor search compared to a linear scan.
We replicated this experiment by:

1. Generating large synthetic datasets of increasing size. Due to computational limitations, we reduced the experimental setup to 100 000, 200 000, 400 000, and 800 000 dataset sizes.
2. Performing approximate nearest-neighbor searches using iSAX indexing.
3. Measuring the rank of the retrieved time series compared to ground truth nearest neighbors.

While we reduced the dataset sizes, all other parameters remained the same as described in the original paper. We also measured the execution time for each dataset setup, which was recorded as [4.5h, 7h, 13.5h, 41.4h] respectively.

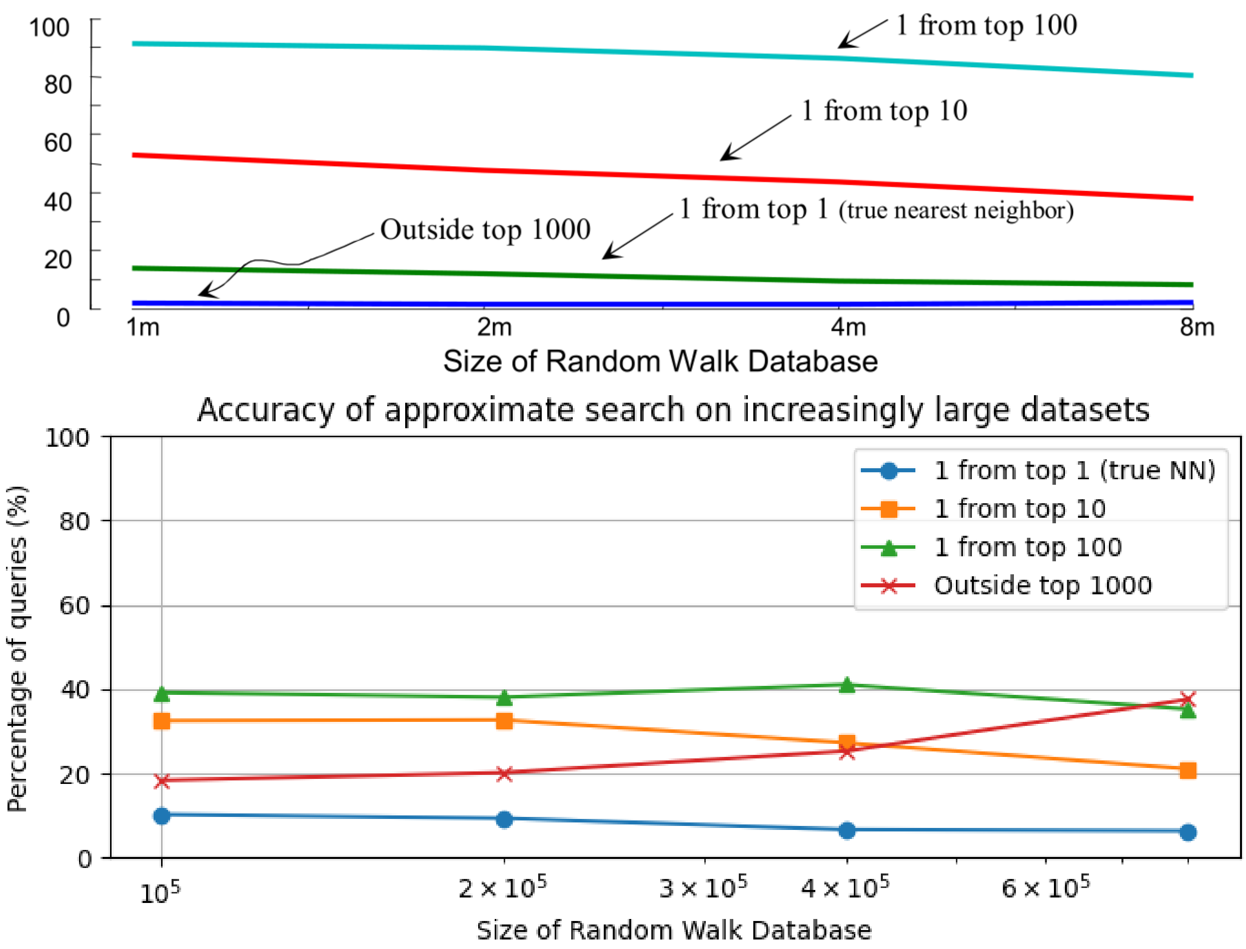

Figure 3: Comparison of Figure 8 from the original paper (top) and our recreated experiment (bottom).

The recreation of the "Indexing Massive Datasets" experiment successfully replicates the key trends observed in the original study but differs in dataset scale and indexing performance. The recreated experiment evaluates dataset sizes different from the original paper's experiment, leading to potential variations in observed trends. The accuracy of approximate search declines as dataset size increases, which is consistent across both experiments. However, in the recreation, the proportion of queries returning the true nearest neighbor (top-1) decreases more sharply, and the percentage of queries falling outside the top 1000 increases noticeably. Despite these differences, the overall trends observed in the recreation align with the claims in the original study, indicating partial success in replication.

To address visualization differences, the recreated experiment was modified to use a logarithmic scale for the x-axis, matching the original study. The differences observed in the recreated experiment could be attributed to several factors. First, the smaller dataset size used in the recreation may result in increased variance in accuracy, as the original study demonstrates that approximate search stabilizes more effectively at larger scales. Second, even though the parameters such as iSAX base cardinality, word length, and threshold for splitting nodes were set to match the original experiment, slight variations in implementation, system architecture, or runtime conditions might still introduce differences in performance. Third, while both experiments use approximate search, subtle differences in search execution, indexing tree depth, or candidate selection strategies may contribute to variations in ranking accuracy, particularly for larger dataset sizes. The observed increase in the percentage of queries falling outside the top 1000 with growing dataset size could be due to the greater number of indexed objects, which increases the difficulty of approximate search in finding high-quality candidates. As dataset size grows, the search space expands, making it harder for approximate methods to retrieve relevant nearest neighbors with high confidence. Additionally, the indexing structure may become less efficient in filtering distant objects, leading to more queries retrieving poor matches.

Overall, while the numerical results of the recreation show some deviations, the general trends and claims of the original study remain valid. The decrease in approximate search accuracy with increasing dataset size, as well as the broader pattern of search ranking degradation, align with the expected behavior. The findings confirm that the iSAX-based indexing approach scales reasonably well, though performance degradation at smaller dataset sizes might be investigated further. Future improvements in parameter tuning and dataset scaling could lead to even stronger consistency with the original findings.

11.4 Computational setup

The implementation was executed in a Python environment using the `numpy`, `scipy`, `requests`, `time`, `shutil`, `os`, `heapq`, and `matplotlib` libraries. All experiments were conducted on a computing system with the following specifications:

- **Processor:** Intel® Core™ i7-14650HX

- **Chipset:** Intel HM770
- **RAM:** 32 GB
- **Storage:** 1000 GB

11.5 Validation of results

To ensure fidelity to the original paper, we compared our output against the expected trends and numerical values derived from the descriptions in the paper. Specifically:

- The breakpoints generated for different cardinalities align with those presented in Table 2.
- The PAA and iSAX representations successfully reproduce the expected symbolic transformations.
- The symbolic distance and query performance trends from Figures 6 and 8 are mostly consistent with the original findings.

Overall, our implementation replicates the experimental methodology, supporting the reproducibility of the study's findings.

12 Conclusion

The iSAX method represents a significant advancement in time series indexing, offering a scalable and efficient solution for large datasets. Through its hierarchical symbolic representation, it successfully bridges the gap between approximate and exact search, making it a valuable tool for data mining applications.

References

[1] Kin-pong Chan and Ada Wai-Chee Fu. "Efficient time series matching by wavelets". In: *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)* (1999), pp. 126–133. URL: https://api.semanticscholar.org/CorpusID:10256010.

[2] Lei Chen and Raymond T. Ng. "On The Marriage of Lp-norms and Edit Distance". In: *Very Large Data Bases Conference.* 2004. URL: https://api.semanticscholar.org/CorpusID:7475159.

[3] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. "Fast subsequence matching in time-series databases". In: *SIGMOD Rec.* 23.2 (May 1994), pp. 419–429. ISSN: 0163-5808. DOI: 10.1145/191843.191925. URL: https://doi.org/10.1145/191843.191925.

[4] Eamonn J. Keogh et al. "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases". In: *Knowledge and Information Systems* 3 (2001), pp. 263–286. URL: https://api.semanticscholar.org/CorpusID:15462747.

[5] Jessica Lin et al. "Experiencing SAX: A Novel Symbolic Representation of Time Series". In: *Data Min. Knowl. Discov.* 15 (Aug. 2007), pp. 107–144. DOI: 10.1007/s10618-007-0064-z.

[6] Dennis Shasha. "Tuning Time Series Queries in Finance: Case Studies and Recommendations". In: *IEEE Data Eng. Bull.* 22 (1999), pp. 40–46. URL: https://api.semanticscholar.org/CorpusID:41032180.

[7] Jin Shieh and Eamonn J. Keogh. "iSAX: indexing and mining terabyte sized time series". In: *Knowledge Discovery and Data Mining.* 2008. URL: https://api.semanticscholar.org/CorpusID:5933532.