Scalable Machine Learning on High-Dimensional Vectors: From Data Series to Deep Network Embeddings

Karima Echihabi
IRDA, Rabat IT Center,
ENSIAS, Mohammed V University
karima.echihabi@gmail.com

Kostas Zoumpatianos
Harvard University
kostas@seas.harvard.edu

Themis Palpanas
LIPADE, Université de Paris &
French University Institute (IUF)
themis@mi.parisdescartes.fr

ABSTRACT

There is an increasingly pressing need, by several applications in diverse domains, for developing techniques able to analyze very large collections of static and streaming sequences (a.k.a. data series), predominantly in real-time. Examples of such applications come from Internet of Things installations, neuroscience, astrophysics, and a multitude of other scientific and application domains that need to apply machine learning techniques for knowledge extraction. It is not unusual for these applications, for which similarity search is a core operation, to involve numbers of data series in the order of hundreds of millions to billions, which are seldom analyzed in their full detail due to their sheer size. Such application requirements have driven the development of novel similarity search methods that can facilitate scalable analytics in this context. At the same time, a host of other methods have been developed for similarity search of high-dimensional vectors in general. All these methods are now becoming increasingly important, because of the growing popularity and size of sequence collections, as well as the growing use of high-dimensional vector representations of a large variety of objects (such as text, multimedia, images, audio and video recordings, graphs, database tables, and others) thanks to deep network embeddings. In this work, we review recent efforts in designing techniques for indexing and analyzing massive collections of data series, and argue that they are the methods of choice even for general high-dimensional vectors. Finally, we discuss the challenges and open research problems in this area.

WIMS 2020, June 30-July 3, 2020, Biarritz, France

ACM ISBN 978-1-4503-7542-9/20/06.
https://doi.org/10.1145/3405962.3405989

CCS CONCEPTS

• **Information systems** → **Proximity search**; **Multidimensional range search**.

KEYWORDS

high-dimensional vectors, data series, time series, embeddings, similarity search, machine learning, deep learning

ACM Reference Format:
Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. 2020. Scalable Machine Learning on High-Dimensional Vectors: From Data Series to Deep Network Embeddings. In *The 10th International Conference on Web Intelligence, Mining and Semantics (WIMS 2020), June 30-July 3, 2020, Biarritz, France.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3405962.3405989

1 INTRODUCTION

Several applications across many diverse domains, such as in Internet of Things (IoT), finance, astrophysics, neuroscience, engineering, multimedia, and others [7, 53, 55, 81], continuously produce big collections of data series[1], which need to be processed and analyzed [11, 12, 35, 38, 46, 62, 65, 77]. Often times, this is part of an exploratory process, where users ask a query, review the results, and then decide what their subsequent queries, or analysis steps should be [55].

The most common type of query that different analysis applications need to answer on these collections of data series is similarity search [26, 27, 53]. Given the continued increase in the rate and volume of data series production, with collections that grow to several terabytes in size [7, 53, 55], several data series indexing and similarity search methods have been developed [26, 27, 31, 32, 54].

In general, similarity search aims at finding objects in a collection that are close to a given query according to some definition of sameness. We note that similarity search is a fundamental operation that goes beyond the data series

[1] A data series, or data sequence, is an ordered sequence of data points. If the ordering dimension is time then we talk about time series, though, series can be ordered over other measures. (e.g., angle in astronomical radial profiles, frequency in infrared spectroscopy, mass in mass spectroscopy, position in genome sequences, etc.).

context: it lies at the core of many critical data science applications. In data integration, it has been used to automate entity resolution [23] and support data discovery [48]. It has powered recommender systems of online billion-dollar enterprises [71] and enabled clustering [13], classification [59] and outlier detection [14] in domains as varied as bioinformatics, computer vision, security, finance and medicine. Similarity search has also been exploited in software engineering [4] to automate API mappings and predict program dependencies and I/O usage, and in cybersecurity to profile network usage and detect intrusions and malware [22].

The similarity search problem has been studied heavily in the past 25 years and will continue to attract attention as massive collections of high-dimensional objects are becoming omnipresent in various domains [7, 55]. Objects can be data series, text, images, audio and video recordings, graphs, database tables or deep network embeddings. Similarity search over high-dimensional objects is often reduced to a k-Nearest Neighbor (k-NN) problem such that the objects are represented using high-dimensional vectors and the (dis)-similarity between them is measured using a distance.

In this work, we review recent efforts in designing techniques for indexing and analyzing massive collections of data series that will enable users to run complex analytics on their data in an interactive fashion, and we argue that they are the methods of choice even for general high-dimensional vector datasets. Finally, we discuss the challenges and open research problems in this area.

2 BACKGROUND AND RELATED WORK

Similarity search has been extensively studied in the past 25 years by different communities. It is an important and challenging problem that is typically modeled as nearest neighbor search in high-dimensional space, where objects are represented as high-dimensional vectors and their (dis)similarity is evaluated using a distance measure such as the Euclidean distance.

Similarity search algorithms can either return exact or approximate answers. Exact methods are expensive while approximate methods sacrifice accuracy to achieve better efficiency. We call the methods that do not provide any guarantees on the results *ng*-approximate, and those that provide guarantees on the approximation error, δ-ϵ-approximate methods, where ϵ is the approximation error and δ, the probability that ϵ will not be exceeded. When $\delta = 1$, a δ-ϵ-approximate method becomes ϵ-approximate, and when $\epsilon = 0$, an ϵ-approximate method becomes exact.

The research community has developed exact [8–10, 19, 28, 33, 73] and approximate [36] similarity search methods for generic high-dimensional vectors[2]. In the past few years, however, we have witnessed a growing interest in the development of approximate methods following two main research trends: (i) LSH-based algorithms [34, 69] that support guarantees, but are relatively slow, and (ii) k-NN graphs [6, 47] and inverted indexes [37, 74], which are relatively fast, but do not provide theoretical guarantees.

The data series community proposed a number of approaches supporting exact search [3, 17, 39, 50, 60, 61, 67], approximate search [20, 21, 40, 66, 70], or both [15, 16, 42–45, 56–58, 64, 72, 75, 76, 79], for this special type of high-dimensional data that exhibits ordered dimensions and correlation between neighboring values.

3 STATE-OF-THE-ART APPROACHES

A large body of data series indexing work has developed on top of the iSAX representation [68]. Figure 1 depicts the lineage of these indexes, along with the corresponding timeline. All these indexes support both Z-normalized[3] and non Z-normalized series, and the same index can answer queries using both the Euclidean and Dynamic Time Warping (DTW) distances (in the way mentioned in [57]), for k-NN and ϵ-range queries [26]. Finally, recent extensions of some of these indexes demonstrate that they can efficiently support approximate similarity search with quality guarantees (deterministic and probabilistic) [27], and that they dominate the state-of-the-art in a variety of settings [24, 26, 27].

Figure 2 presents a (non-exhaustive) taxonomy of similarity search methods based on the type of guarantees they provide (methods with multiple types of guarantees are included in more than one leaf of the taxonomy). In this figure, we list methods that have been developed for data series and general high-dimensional vectors alike.

Methods that provide no guarantees are categorized under ng-approximate, and they include tree-based indexes (ADS+ [79], DSTree [72], HD-index [5], iSAX2+ [16], SFA [64], Flann [51]), graph-based techniques (HNSW [47], NSG [29]), quantization-based approaches (IMI [6, 30], CK-Means [52]) and scans (VA+file [28]).

We call probabilistic the δ-ϵ-approximate methods when $\delta < 1$. These include the Mtree [18], techniques from the LSH family (QALSH [34], SRS [69]) and the extensions to the exact data series techniques ADS+, DSTree, iSAX2+ and VA+file, proposed in [27]. When $\delta = 1$ we have the ϵ-approximate methods which span the Mtree and the extensions.

[2] An exhaustive survey can be found in [63].

[3] Z-normalization transforms a series so that it has a mean value of zero, and a standard deviation of one. This allows similarity search to be effective, irrespective of shifting (i.e., offset translation) and scaling[41]. Therefore, similarity search can return results with similar trends, but different absolute values. Moreover, minimizing the Euclidean distance on Z-normalized data is equivalent to maximizing their Pearson's correlation coefficient [49]. For these reasons, Z-normalization is extensively used in both the literature [26, 27, 80] and in practice [7, 55].

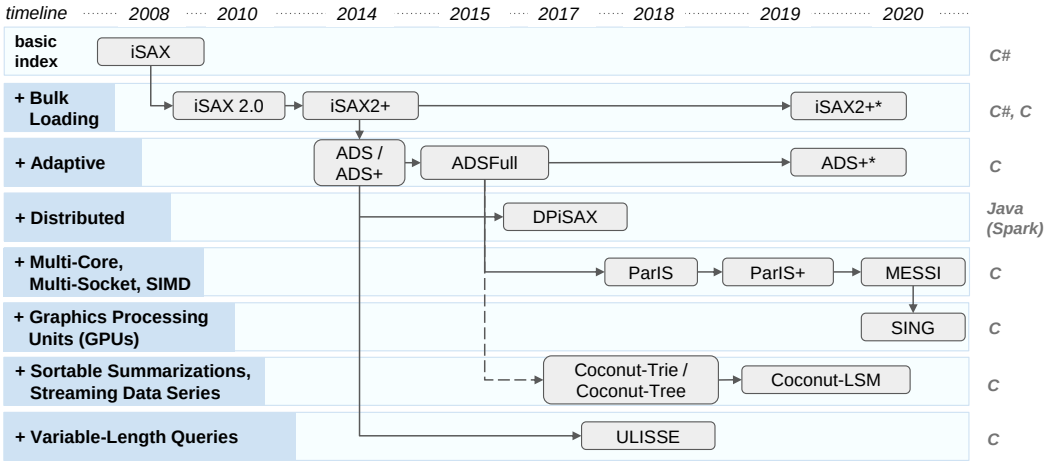

Figure 1: Lineage of the iSAX family of indexes. Timeline is depicted on the top; implementation languages are marked on the right. Solid arrows denote inheritance of the index design; dashed arrows denote inheritance of some of the design features; the two new versions of iSAX2+ and ADS+ marked with an asterisk support approximate similarity search with deterministic and probabilistic quality guarantees. Source code available by following the links in the corresponding papers.

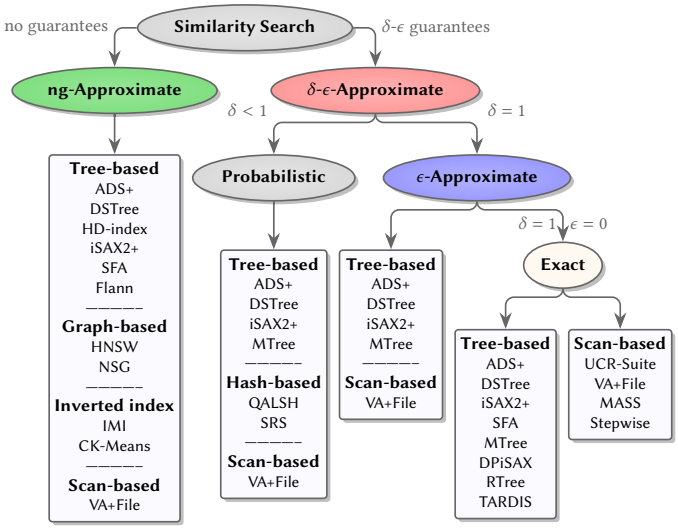

Figure 2: Taxonomy of similarity search methods.

Setting $\delta = 1$ and $\epsilon = 0$, we get the exact methods including tree-based indexes (ADS+ [79], DSTree [72], DPiSAX [75], iSAX2+ [16], MTree [19], RTree [8], SFA [64], TARDIS [78]) and the scan-based techniques (MASS [50], Stepwise [39], UCR-Suite [61], VA+File [28]).

4 DISCUSSION

We recently conducted the two most extensive and comprehensive experimental evaluations in this area [26, 27], including techniques from both the data series and high-dimensional data communities, which had never been considered together. The results indicate that approaches designed for data series outperform the state-of-the-art approximate and exact high-dimensional vector techniques.

These results have far-reaching fundamental and practical implications. They demonstrate that it is possible to design efficient high-dimensional vector similarity search algorithms with theoretical guarantees on the quality of the answers, and thus, offer a more promising alternative to the two current trends in the literature. This finding paves the way for very exciting new developments which will lead to efficient solutions that can support critical analytical tasks such as brain seizure detection, cyber-attack prevention, transportation management and data cleaning automation.

Our first study [26] assessed the efficiency and footprint of exact data series similarity search algorithms against popular techniques from the high-dimensional community, such as the R*-tree [8], the M-tree [19] and the VA+file [28]. In the approximate similarity search literature, experimental evaluations ignore the answering capabilities of data series methods. Our second experimental evaluation [27] is the first study that fills this gap comparing the efficiency, accuracy and footprint of approximate data series approaches to state-of-the-art techniques designed for high-dimensional vectors such as SRS [69], QALSH [34], IMI [6] and HNSW [47]. Since the existing data series techniques supported either exact or *ng*-approximate similarity search, we extended the best

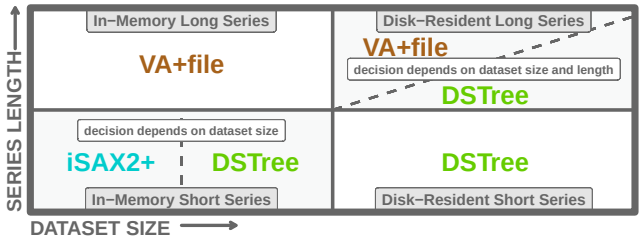

Figure 3: Exact search recommendations [26]

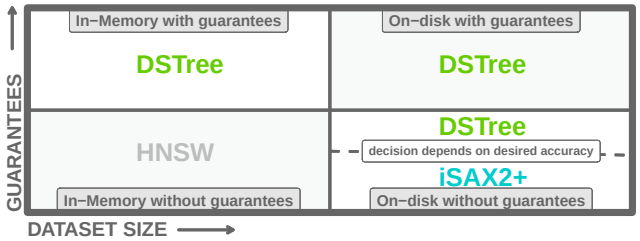

Figure 4: Approximate search recommendations [27]

techniques per [26] using ideas that were proposed 20 years ago for metric trees [18], and have not been exploited until now by any other technique.

The main contributions of these studies are the following.

(1) A deeper understanding of data series exact similarity search. We provide a formal problem definition unifying conflicting terminology from different research communities, and we present a taxonomy that classifies similarity search methods based on the search quality guarantees.

We argue that access path selection for data series similarity search is an optimization problem that depends on a variety of factors including hardware, query pruning ratio, data characteristics, the accuracy of a summarization and the efficacy of the clustering provided by an index. To help users decide the best approach for their problem, we issue a set of recommendations (Figure 3) for a particular hardware and query workload. In this scenario, the VA+file [28] is particularly well-suited for long series in-memory while for shorter series, the DSTree [72] is the best contender on disk, and iSAX2+ [16] is the winner in-memory.

We present an elaborate discussion and pinpoint the approaches that would benefit the most from modern hardware.

(2) A deeper understanding of high-dimensional vector approximate similarity search. We pinpoint the weaknesses and the strengths of the different techniques sharing insights that have never been published in the literature. For instance, LSH techniques such as SRS and QALSH exploit both δ and ϵ to tune the efficiency/accuracy tradeoff. We show that their performance is still inadequate, because a low ϵ and a high δ can still lead to inaccurate answers and low values of ϵ and δ can still result in slow execution. The *ng*-approximate methods IMI and HNSW provide better efficiency but they suffer from three major limitations: (a) they have no guarantees; (b) they are very difficult to tune; and (c) their speed-accuracy tradeoff is not determined only at query time, but also during index building. Choosing the best approach to answer an approximate similarity search query depends on a variety of factors including the accuracy desired, the dataset characteristics, the size of the query workload, the presence of an existing index and the hardware. Figure 4 illustrates a decision matrix that recommends the best technique to use for answering a query workload using an existing index. Overall, DSTree is the best performer, with the exceptions of ng-approximate queries, where iSAX2+ also exhibits excellent performance, and of in-memory datasets, where HNSW is the overall winner. Accounting for index construction time as well, DSTree becomes the method of choice across the board, except for small workloads, where iSAX2+ wins.

(3) A new set of approximate techniques for high-dimensional vector similarity search that outperform the state-of-the-art. Our proposed techniques are the clear winners for δ-ϵ-approximate similarity search for both in-memory and disk based data, while they are the only viable solution for on-disk data. These techniques are also the fastest at indexing and have the lowest footprint. The only scenario where our techniques are outperformed by another method is for in-memory *ng*-approximate search where HNSW [47] is the best contender. However, this kNN graph-based method has very high footprint, is difficult to tune and can return incomplete results.

(4) A public archive which would serve as a the stepping stone for a much-needed benchmark. In both studies, all methods are evaluated under a unified framework to prevent implementation bias. We used the most efficient C/C++ implementations available for all approaches, and developed from scratch in C the ones that were only implemented in other programming languages, leading to new implementations that are considerably faster than the original ones. We share a public archive containing all source codes, datasets, queries and results [1, 2].

5 CONCLUSIONS AND OUTLOOK

During the last years we have witnessed a flurry of activity related to data series analytics, for which a core operation is similarity search. Therefore, several similarity search methods that can facilitate scalable analytics in this context were developed. At the same time, a host of other methods have been developed for similarity search of high-dimensional vectors in general. All these methods are now becoming increasingly important, because of the growing popularity and size of sequence collections, as well as the growing use of

high-dimensional vector representations of a large variety of objects, thanks to deep network embeddings.

In this work, we review recent efforts in designing techniques for indexing and analyzing massive collections of data series, and argue that they are the methods of choice even for general high-dimensional vectors. The insights gained from two experimental evaluation studies on similarity search methods, regarding the strengths and weaknesses of existing approaches, indicate that further improvements in the design and performance of similarity search methods are possible. A key future direction is also the development of methods that will support progressive query answering with probability guarantees [25, 31, 32] that could be consumed by human analysts and other analysis algorithms, which will enable interactive data exploration on massive collections of data series and high-dimensional vectors alike.

REFERENCES

[1] 2018. Lernaean Hydra Archive. http://www.mi.parisdescartes.fr/~themisp/dsseval/.
[2] 2019. Lernaean Hydra Archive II. http://www.mi.parisdescartes.fr/~themisp/dsseval2/.
[3] Rakesh Agrawal, Christos Faloutsos, and Arun N. Swami. 1993. Efficient Similarity Search In Sequence Databases. In *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*.
[4] Uri Alon, Meital Zilberstein, Omer Levy, and Eran Yahav. 2019. Code2vec: Learning Distributed Representations of Code. 3, POPL (2019).
[5] Akhil Arora, Sakshi Sinha, Piyush Kumar, and Arnab Bhattacharya. 2018. HD-index: Pushing the Scalability-accuracy Boundary for Approximate kNN Search in High-dimensional Spaces. *PVLDB* 11, 8 (2018), 906–919.
[6] A. Babenko and V. Lempitsky. 2015. The Inverted Multi-Index. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 6 (2015), 1247–1260.
[7] Anthony J. Bagnall, Richard L. Cole, Themis Palpanas, and Konstantinos Zoumpatianos. 9(7), 2019. Data Series Management. *Dagstuhl Reports* (9(7), 2019).
[8] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. 1990. The R*-tree: an efficient and robust access method for points and rectangles. In *International Conference on Management of Data*. ACM, 322–331.
[9] Jon Louis Bentley. 1975. Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM* 18, 9 (Sept. 1975), 509–517.
[10] Stefan Berchtold, Daniel A. Keim, and Hans-Peter Kriegel. 1996. The X-tree: An Index Structure for High-Dimensional Data. In *VLDB*.
[11] Paul Boniol, Michele Linardi, Federico Roncallo, and Themis Palpanas. 2020. Automated Anomaly Detection in Large Sequences. In *ICDE*.
[12] Paul Boniol and Themis Palpanas. 2020. Series2Graph: Graph-based Subsequence Anomaly Detection for Time Series. *PVLDB* (2020).
[13] Sébastien Bubeck and Ulrike von Luxburg. 2009. Nearest Neighbor Clustering: A Baseline Method for Consistent Clustering with Arbitrary Objective Functions. *JMLR* 10 (2009).
[14] Simon Byers and Adrian E. Raftery. 1998. Nearest-Neighbor Clutter Removal for Estimating Features in Spatial Point Processes. *JASA* 93, 442 (1998).
[15] Alessandro Camerra, Themis Palpanas, Jin Shieh, and Eamonn J. Keogh. 2010. iSAX 2.0: Indexing and Mining One Billion Time Series.. In *ICDM*. IEEE Computer Society, 58–67.
[16] Alessandro Camerra, Jin Shieh, Themis Palpanas, Thanawin Rakthanmanon, and Eamonn J. Keogh. 2014. Beyond one billion time series: indexing and mining very large time series collections with iSAX2+. *Knowl. Inf. Syst.* 39, 1 (2014), 123–151.
[17] Georgios Chatzigeorgakidis, Dimitrios Skoutas, Kostas Patroumpas, Themis Palpanas, Spiros Athanasiou, and Spiros Skiadopoulos. 2019. Local Pair and Bundle Discovery over Co-Evolving Time Series. In *SSTD*.
[18] Paolo Ciaccia and Marco Patella. 2000. PAC Nearest Neighbor Queries: Approximate and Controlled Search in High-Dimensional and Metric Spaces. In *ICDE*. 244–255.
[19] Paolo Ciaccia, Marco Patella, and Pavel Zezula. 1997. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB'97)*. 426–435.
[20] Richard Cole, Dennis E. Shasha, and Xiaojian Zhao. 2005. Fast window correlations over uncooperative time series. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 743–749.
[21] Michele Dallachiesa, Themis Palpanas, and Ihab F. Ilyas. 2014. Top-k Nearest Neighbor Search in Uncertain Data Series. *PVLDB* 8, 1 (2014), 13–24.
[22] Sumeet Dua and Xian Du. 2011. *Data Mining and Machine Learning in Cybersecurity* (1st ed.). Auerbach Publications, USA.
[23] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed Representations of Tuples for Entity Resolution. *VLDBJ* 11, 11 (2018).
[24] Karima Echihabi. 2019. Truly Scalable Data Series Similarity Search. In *Proceedings of the VLDB 2019 PhD Workshop, co-located with the 45th International Conference on Very Large Databases (VLDB 2019)*.
[25] Karima Echihabi. 2020. High-Dimensional Vector Similarity Search: From Time Series to Deep Network Embeddings. In *SIGMOD*.
[26] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2018. The Lernaean Hydra of Data Series Similarity Search: An Experimental Evaluation of the State of the Art. *PVLDB* (2018).
[27] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2019. Return of the Lernaean Hydra: Experimental Evaluation of Data Series Approximate Similarity Search. *PVLDB* (2019).
[28] Hakan Ferhatosmanoglu, Ertem Tuncel, Divyakant Agrawal, and Amr El Abbadi. 2000. Vector Approximation Based Indexing for Non-uniform High Dimensional Data Sets. In *CIKM*.
[29] Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. 2019. Fast Approximate Nearest Neighbor Search with the Navigating Spreading-out Graph. *PVLDB* 12, 5 (2019), 461–474.
[30] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2014. Optimized Product Quantization. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 4 (April 2014), 744–755. https://doi.org/10.1109/TPAMI.2013.240
[31] Anna Gogolou, Theophanis Tsandilas, Karima Echihabi, Themis Palpanas, and Anastasia Bezerianos. 2020. Data Series Progressive Similarity Search with Probabilistic Quality Guarantees. In *SIGMOD*.
[32] Anna Gogolou, Theophanis Tsandilas, Themis Palpanas, and Anastasia Bezerianos. 2019. Progressive Similarity Search on Time Series Data. In *Proceedings of the Workshops of the EDBT/ICDT 2019 Joint Conference*.
[33] Antonin Guttman. 1984. R-Trees: A Dynamic Index Structure for Spatial Searching. In *SIGMOD*.
[34] Qiang Huang, Jianlin Feng, Yikai Zhang, Qiong Fang, and Wilfred Ng. 2015. Query-aware Locality-sensitive Hashing for Approximate Nearest Neighbor Search. *PVLDB* 9, 1 (2015), 1–12.
[35] Pablo Huijse, Pablo A Estevez, Pavlos Protopapas, Jose C Principe, and Pablo Zegers. 2014. Computational intelligence challenges and

applications on large-scale astronomical time series databases. *CIM* (2014).
[36] Piotr Indyk and Rajeev Motwani. 1998. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *STOC*.
[37] H. Jegou, M. Douze, and C. Schmid. 2011. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1 (2011), 117–128.
[38] Kunio Kashino, Gavin Smith, and Hiroshi Murase. 1999. Time-series active search for quick retrieval of audio and video. In *ICASSP*.
[39] Shrikant Kashyap and Panagiotis Karras. 2011. Scalable kNN search on vertically stored time series.. In *KDD*. ACM, 1334–1342.
[40] Eamonn Keogh and Padhraic Smyth. 1997. A Probabilistic Approach to Fast Pattern Matching in Time Series Databases. In *KDD*. AAAI Press, 24–30.
[41] Eamonn J. Keogh and Shruti Kasetty. 2003. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. *DAMI* (2003).
[42] Haridimos Kondylakis, Niv Dayan, Kostas Zoumpatianos, and Themis Palpanas. 2018. Coconut: A Scalable Bottom-Up Approach for Building Data Series Indexes. *PVLDB* 11, 6 (2018), 677–690.
[43] Haridimos Kondylakis, Niv Dayan, Kostas Zoumpatianos, and Themis Palpanas. 2019. Coconut: Sortable Summarizations for Scalable Indexes over Static and Streaming Data Series. *VLDBJ* 28, 6 (2019), 847–869.
[44] Michele Linardi and Themis Palpanas. 2018. Scalable, Variable-length Similarity Search in Data Series: The ULISSE Approach. *PVLDB* 11, 13 (2018), 2236–2248.
[45] Michele Linardi and Themis Palpanas. 2018. ULISSE: ULtra compact Index for Variable-Length Similarity SEarch in Data Series. In *ICDE*.
[46] Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn J. Keogh. 2020. Matrix Profile Goes MAD: Variable-Length Motif And Discord Discovery in Data Series. In *DAMI*.
[47] Yury A. Malkov and D. A. Yashunin. 2016. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. *CoRR* abs/1603.09320 (2016).
[48] Renée J. Miller. 2018. Open Data Integration. *PVLDB* 11, 12 (2018), 2130–2139.
[49] Abdullah Mueen, Suman Nath, and Jie Liu. 2010. Fast approximate correlation for massive time-series data. In *SIGMOD*.
[50] Abdullah Mueen, Yan Zhu, Michael Yeh, Kaveh Kamgar, Krishnamurthy Viswanathan, Chetan Gupta, and Eamonn Keogh. 2017. The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance.
[51] Marius Muja and David G. Lowe. 2009. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP International Conference on Computer Vision Theory and Applications*. 331–340.
[52] Mohammad Norouzi and David J. Fleet. 2013. Cartesian K-Means. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*. 3017–3024.
[53] Themis Palpanas. 2015. Data Series Management: The Road to Big Sequence Analytics. *SIGMOD Record* (2015).
[54] Themis Palpanas. 2020. Evolution of a Data Series Index. *CCIS* 1197 (2020).
[55] Themis Palpanas and Volker Beckmann. 2019. Report on the First and Second Interdisciplinary Time Series Analysis Workshop (ITISA). *SIGMOD Rec.* 48, 3 (2019).
[56] Botao Peng, Panagiota Fatourou, and Themis Palpanas. 2018. ParIS: The Next Destination for Fast Data Series Indexing and Query Answering. *IEEE BigData* (2018).
[57] Botao Peng, Themis Palpanas, and Panagiota Fatourou. 2020. MESSI: In-Memory Data Series Indexing. In *ICDE*.
[58] Botao Peng, Themis Palpanas, and Panagiota Fatourou. 2020. ParIS+: Data Series Indexing on Multi-core Architectures. *TKDE* (2020).
[59] François Petitjean, Germain Forestier, Geoffrey I. Webb, Ann E. Nicholson, Yanping Chen, and Eamonn J. Keogh. 2014. Dynamic Time Warping Averaging of Time Series Allows Faster and More Accurate Classification. In *ICDM*.
[60] Davood Rafiei. 1999. On Similarity-Based Queries for Time Series Data. In *ICDE*. 410–417.
[61] Thanawin Rakthanmanon, Bilson J. L. Campana, Abdullah Mueen, Gustavo E. A. P. A. Batista, M. Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn J. Keogh. 2012. Searching and mining trillions of time series subsequences under dynamic time warping.. In *KDD*. ACM, 262–270.
[62] Usman Raza, Alessandro Camerra, Amy L Murphy, Themis Palpanas, and Gian Pietro Picco. 2015. Practical data prediction for real-world wireless sensor networks. *TKDE* (2015).
[63] Hanan Samet. 2005. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers Inc.
[64] Patrick Schäfer and Mikael Högqvist. 2012. SFA: A Symbolic Fourier Approximation and Index for Similarity Search in High Dimensional Datasets. In *EDBT*.
[65] Dennis Shasha. 1999. Tuning time series queries in finance: Case studies and recommendations. *IEEE Data Eng. Bull.* (1999).
[66] H. Shatkay and S. B. Zdonik. 1996. Approximate queries and representations for large data sequences. In *ICDE*. 536–545.
[67] Jin Shieh and Eamonn Keogh. 2008. iSAX: Indexing and Mining Terabyte Sized Time Series. In *SIGKDD*. ACM, 623–631.
[68] Jin Shieh and Eamonn Keogh. 2008. iSAX: Indexing and Mining Terabyte Sized Time Series. In *KDD*.
[69] Yifang Sun, Wei Wang, Jianbin Qin, Ying Zhang, and Xuemin Lin. 2014. SRS: Solving c-approximate Nearest Neighbor Queries in High Dimensional Euclidean Space with a Tiny Index. *PVLDB* 8, 1 (2014).
[70] Changzhou Wang and Xiaoyang Sean Wang. 2000. Supporting content-based searches on time series via approximation. In *SSDBM*. 69–81.
[71] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-Scale Commodity Embedding for E-Commerce Recommendation in Alibaba. In *KDD*.
[72] Yang Wang, Peng Wang, Jian Pei, Wei Wang, and Sheng Huang. 2013. A Data-adaptive and Dynamic Segmentation Index for Whole Matching on Time Series. *PVLDB* 6, 10 (2013), 793–804.
[73] Roger Weber, Hans-Jörg Schek, and Stephen Blott. 1998. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases*. 194–205.
[74] Yan Xia, Kaiming He, Fang Wen, and Jian Sun. 2013. Joint Inverted Indexing. *2013 IEEE International Conference on Computer Vision* (2013), 3416–3423.
[75] D. E. Yagoubi, R. Akbarinia, F. Masseglia, and T. Palpanas. 2017. DPiSAX: Massively Distributed Partitioned iSAX. In *ICDM*. 1135–1140.
[76] Djamel-Edine Yagoubi, Reza Akbarinia, Florent Masseglia, and Themis Palpanas. 2020. Massively Distributed Time Series Indexing and Querying. *IEEE Trans. Knowl. Data Eng.* 32, 1 (2020), 108–120.
[77] Lexiang Ye and Eamonn Keogh. 2009. Time series shapelets: a new primitive for data mining. In *SIGKDD*. ACM.
[78] L. Zhang, N. Alghamdi, M. Y. Eltabakh, and E. A. Rundensteiner. 2019. TARDIS: Distributed Indexing Framework for Big Time Series Data. In *ICDE*.
[79] Kostas Zoumpatianos, Stratos Idreos, and Themis Palpanas. 2016. ADS: the adaptive data series index. *VLDBJ* 25, 6 (2016), 843–866.
[80] Kostas Zoumpatianos, Yin Lou, Ioana Ileana, Themis Palpanas, and Johannes Gehrke. 2018. Generating data series query workloads. *VLDB J.* (2018).
[81] Kostas Zoumpatianos and Themis Palpanas. 2018. Data Series Management: Fulfilling the Need for Big Sequence Analytics. In *ICDE*.