EuroVis Workshop on Visual Analytics (2022)
M. Angelini and J. Bernard (Editors)

Multivariate Time Series Retrieval with Symbolic Aggregate Approximation, Regular Expression, and Query Expansion

Y. Yu[1,2], T. Becker[1,2] and M. Behrisch[1]

[1]Utrecht University, Netherlands
[2]IAV GmbH Ingenieurgesellschaft Auto und Verkehr, Germany

Figure 1: ***Pipeline of the proposed method:*** *1) Stringify the time series to allow text retrieval methods; 2) zipping tracks to enable multivariate retrieval; 3) extract regex from the stringified query; 4) expand the query to tackle distortions; 5) search with regex engine.*

Abstract

We present `SAXRegEx`, a method for pattern search in multivariate time series in the presence of various distortions, such as duration variation, warping, and time delay between signals. For example, in the automotive industry, calibration engineers spontaneously search for event-induced patterns in fresh measurements under time pressure. Current methods do not sufficiently address duration (horizontal along the time axis) scaling and inter-track time delay. One reason is that it can be overwhelmingly complex to consider scaling and warping jointly and analyze temporal dynamics and attribute interrelation simultaneously. `SAXRegEx` meets this challenge with a novel symbolic representation modeling adapted to handle time series with multiple tracks. We employ methods from text retrieval, i.e., regular expression matching, to perform a pattern retrieval and develop a novel query expansion algorithm to deal flexibly with pattern distortions. Experiments show the effectiveness of our approach, especially in the presence of such distortions, and its efficiency surpassing the state-of-the-art methods. While we design the method primarily for automotive data, it is well transferable to other domains.

CCS Concepts

• ***Mathematics of computing*** → *Time series analysis;* • ***Information systems*** → *Query representation;*

1. Introduction

When a car starts, a series of events are recorded for analysis and control. These event sequences are used in various use cases. Engine engineers await a signal to pinpoint the best entry point of their novel algorithm for a smoother air-fuel ratio control. Transmission engineers trace gear change intervals to analyze the moments on different shafts for less traction loss. Noise Vibration Harshness (NVH) engineers hunt for periods when some cylinders are deactivated or reactivated for cars with variable displacement to study the abrupt vibration during these periods. These use cases are all backed up by techniques for pattern search in time series.

Locating sub-sequences in time series similar to the given query is a frequent prerequisite for further data processing. Univariate retrieval with little distortion is efficiently solved with numerous time series indexing techniques [AJ02, GDK*21]. Univariate case with certain distortion types, especially time shifts and sometimes horizontal scaling along the time axis, is also addressed [Keo97, JL18a, SSA*18]. Recently, multivariate cases have received increasing attention [MJE15, LL18]. Multivariate distortion-invariant time series retrieval, however, remains largely untouched.

Our target signals constantly vary in duration and are often distributed in several tracks with significant inter-track time delay. Horizontal length scaling and time shifts within a track can be regarded as heterogeneous horizontal resampling. The dummy data in Figure 2 illustrate the distortions. These distortions are also likely to plague the patterns in data from other domains. Heterogeneous horizontal scaling is hard to capture, due to the de facto preprocessing step with a sliding window, usually assuming a fixed pattern length silently. Inter-track time shifts are even more challenging due to the complexity during simultaneous consideration of the temporal dynamics and the interrelation between attributes.

We propose an easy-to-implement method, `SAXRegEx`, based on symbolic representation to enable text retrieval methods and regular expressions as our search engine. Query expansion is designed

DOI: 10.2312/eurova.20221081

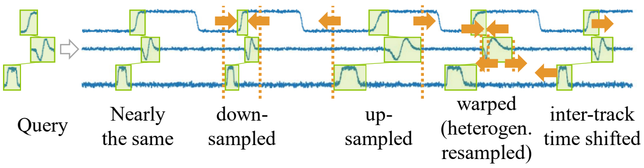

Figure 2: ***Distorted patterns:*** *the query on the left is searched for in the multivariate time series on the right. Target patterns in the time series exhibit distortion in various forms.*

to deal with the mentioned distortions. Our method shows comparable accuracy to state-of-the-art techniques on datasets without these distortions and better performance in multivariate datasets with mentioned distortions. Additionally, `SAXRegEx` outperforms the state-of-the-art in terms of speed.

2. Related Work

Research on time series retrieval mainly focused on two directions: first, novel similarity measures between two time series that better describe the notion of similarity (accuracy) [SSA*18, JL18a]; Second, indexing techniques focusing on efficiency [LS95, GDK*21, PFP21]. Usually, sliding windows are assumed for preprocessing. Our work breaks the convention by employing neither a sliding window approach nor an explicit measure of similarity.

Besides random noise, patterns in the time series can have various distortion forms. Horizontal translation along the time axis is trivial; horizontal scaling is only addressed in a few works, such as [Keo97, MA18]; vertical translation and scaling on the value/amplitude axis is often handled by normalizing time series windows [LPH*20, GDK*21]. Warping is captured by a variety of "elastic" distances [CN04, AVG13], especially Dynamic Time Warping (DTW) [BC94, MJE15, JL18a]. In multivariate cases, most methods handle the temporal dynamics and interrelation between tracks independently, assuming that the tracks are synchronized. For instance, DTW_I calculates distance profiles for each track individually and merges them subsequently [MJE15]; Locality-Sensitive Hashing (LSH) merges tracks followed by univariate processing [CLL*19]. All these approaches assume that the pattern is synchronized in all tracks. Driven by our domain problems, we target this problem without this assumption and scan the temporal dynamics with inter-track relation jointly.

Recently, machine learning penetrates time series retrieval [JL18b, LXZ19, LPH*20]. They do not suit, however, highly dynamic or user-driven retrieval cases, where the queries and the datasets change rapidly. In our case, engineers start the search spontaneously and wish an immediate answer.

Regarding accuracy, it is accepted that no similarity measure accounts for the human notion of similarity [MM16]; therefore, different datasets favor different similarity measures. Some works propose an active learning strategy to adapt the similarity measure [KP98, LPH*20, YKBB21]. The extensive benchmark in [BLB*17] shows that DTW enjoys top ranking accuracy among 20 methods for time series classification. As for speed, Mueen's Algorithm for Similarity Search (MASS) is so far the fastest similarity search algorithm [YZU*16, MAA*21]. We, consequently, include both for benchmarking `SAXRegEx` (Section 4).

`SAXRegEx`'s basic concept, regex-based search in symbolic-encoded time series, resembles SSTS [JDDH19]. In comparison, `SAXRegEx`'s contributions lie in the ability to handle multivariate data and various distortions.

3. Method

`SAXRegEx` consists of five steps illustrated in Figure 3. First, it encodes with SAX each track in the query and in the time series. It reduces data volume, smooths curves, and enables methods for text retrieval (details in Appendix A). Next, `SAXRegEx` zips tracks in the query and in the time series. This step enables simultaneous processing of multiple tracks through a single regex. Here, the word "zip" does not refer to the lossless compression but rather step-wise merging of sequences. In the third step, repetitive symbol groups are merged through regex quantifiers, which exploits SAX's "numerosity reduction" property [LKWL07]. The penultimate step conducts query expansion to cope with distortions. As shown in Figure 3, it adds a tolerance band to the query by allowing character classes rather than character instances; thereupon, it makes the query elastic along the time axis by substituting the fixed quantifier with ranges. Finally, `SAXRegEx` searches for the query regex in the time series string and reconstructs the pattern in each track by a fine regex matching in the predicted intervals. In the following sections, we describe Step 3 and Step 4 in detail with examples.

3.1. Regex Extraction

The first two steps of our processing pipeline typically leave repetitive symbols in the text sequences. They can be bundled with quantifiers. For instance, the string `ACCCCGGGBAAA` can be rewritten as `AC{4}G{3}BA{3}`. As we will see later, these quantifies also enable horizontal scaling invariant search.

Generally, regex is meant for exact search instead of "fuzzy" search like pattern search in time series. It can only match patterns strictly satisfying the restrictions imposed by the regex. To allow fuzzy search, we add a tolerance band to the query with character classes analogous to the L^∞ norm. For instance, the previously encoded query `AC{4}G{3}BA{3}` can be further augmented as `[A-B][B-D]{4}[F-H]{3}[A-C][A-B]{3}`. In this example, we allow a tolerance band of two symbols.

3.2. Query Expansion

We conduct distortion-specific query expansion to address the necessary retrieval invariances required in our use cases, particularly heterogeneous horizontal scaling and inter-track time shifts.

For heterogeneous horizontal scaling, we use value ranges rather than fixed regex quantifiers. Pertaining to our running example, we can further augment our query to `[A-B]{0-2}[B-D]{2-8}[F-H]{1-6}[A-C]{0-2}[A-B]{1-6}`. This modification captures similar patterns with a half to double duration. Moreover, it allows different fragments in the pattern to have different scaling factors, thus capturing complex heterogeneity and warping.

For inter-track time shifts, we use a different alphabet per track and zip the tracks to a single string. As shown in Figure 3 Step

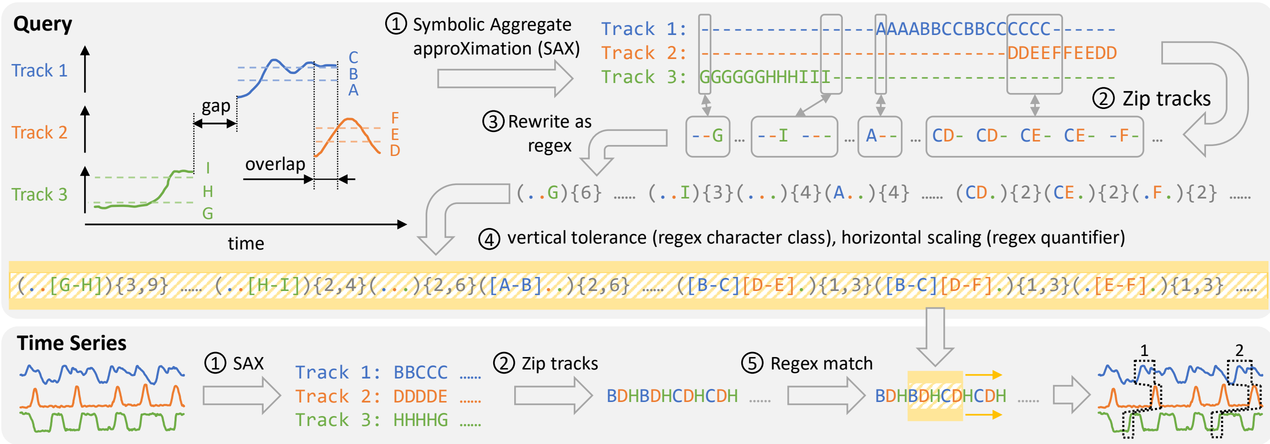

Figure 3: ***Pipeline:*** *1) convert with SAX each track with a different alphabet to enable text retrieval methods; 2) merge tracks in the query and in the time series by zipping to process multiple tracks with a single regex, leave wildcards "." for "gaps" to capture inter-track time shifts; 3) extract regex from the query string by combining repetitive symbol groups; 4) manipulate the regex with character classes and quantifiers to deal with distortions; 5) search for the query regex in the time series string and reconstruct the shape in each track.*

2, we zip the query while taking time shifts between the shapes in different tracks into consideration. We can add this flexibility by introducing the wildcard character "." for "gaps". This way, the query ignores the potentially interfering context in each track and focuses only on the given shape in the track. As a result, the shape in a track within the query does not have to span the whole query length. Note in Figure 3, the term `(...){2,6}`, which captures the elastic gap between the shapes in Track 3 and in Track 1.

4. Evaluation

We evaluated the accuracy and speed of `SAXRegEx` with eight labeled datasets, among which two contain patterns with heterogeneous horizontal scaling or inter-track time shifts. We chose four state-of-the-art methods for our comparative evaluation of `SAXRegEx`: 1) correlation: one of the standard similarity measures; 2) DTW: the most popular elastic distance measure for time series with "hard-to-beat" accuracy for time series classification [BLB*17]; 3) Euclidean Distance (ED): specifically MASS, the so-far fastest similarity search tool [YZU*16, MAA*21]; 4) SAX: our baseline time series representation, together with its similarity measure [LKWL07], can be used for similarity search. Details of the benchmark methods are in Appendix C and Appendix D.

All experiments were conducted on a standard laptop running on 64-bit Windows 10 Enterprise with Intel i7-8650U CPU, 16GB RAM, and 1TB HDD. Details of the datasets, hardware and software setup are in Appendix B and Appendix C, respectively.

4.1. Accuracy Benchmark

We compared the performance of all methods quantitatively with standard metrics and qualitatively through a visual inspection. Besides the standard metrics accuracy, balanced accuracy, precision, recall, and F1-score, which regard the problem as the binary classification for each time step (inside/outside a pattern), we want to introduce the metric mean Average Precision (mAP) from object detection in computer vision from a segment-wise perspective. We have chosen 30% and 50% as the Intersection over Union (IoU)-threshold required by this metric. They are denoted mAP30 and mAP50, respectively. For details, please refer to Appendix C. The table in Figure 4 (complete version in Appendix E) depicts our five evaluation metrics for all methods on various datasets. The result suggests that different data favor different methods and no method constantly outperforms the other, confirming the finding in [MM16]. However, `SAXRegEx` can better capture heterogeneous horizontal scaling and inter-track time shifts. We can observe this specifically on the last two datasets in Figure 4 and Figure E.7.

We have plotted the predictions and ground truth labels with all methods on all publishable datasets in Appendix D, together with the description of the plots. Figure 5 shows an example. From the visual inspection, we can infer pitfalls of the benchmark methods. Firstly, the benchmark methods spread focus evenly over all time steps rather than critical transitions (usually large ramps). This leads to the false positive Prediction 5 with ED, which misses an upward ramp at the beginning in the first track and a downward ramp at the end in the second track. Next, the benchmark methods take the context of the pattern shape in each track into account (please note the difference between the query for `SAXRegEx` and the benchmark methods). Consequently, the search can be misled by a changing context, as implied by the missing ground truth label 2 for ED and DTW. In this case, the target pattern contains a distracting plateau in each track. Finally, while DTW well addresses the time shifts within a track, as proven by the performance on the CAN 1 dataset in Figure 4 and Figure D.5, it has a hard time when it comes to time shifts between tracks, as indicated by Prediction 2, 3 and 4 for DTW. In the presence of such distortions, `SAXRegEx` outperforms the benchmarks and fulfills our needs.

4.2. Speed Benchmark

We calculated the elapsed time of all methods on all datasets. Every experiment is repeated five times. The complete result can be found

		Prec.	Recall	F1	mAP30	mAP50
Cable Cutter	Corr.	0.79	0.86	**0.82**	**0.89**	**0.56**
	DTW	0.48	0.98	0.64	0.84	0.19
	ED	0.36	0.52	0.43	0.27	0.18
	SAX	0.74	0.71	0.72	0.69	0.31
	SRe	0.55	0.87	0.67	**0.89**	0.55
Deep Valve	Corr.	0.88	0.47	0.61	0.82	0.29
	DTW	0.46	0.94	0.62	0.46	0.19
	ED	0.88	0.53	0.67	**0.85**	0.28
	SAX	0.87	0.93	**0.90**	0.73	**0.66**
	SRe	0.62	0.84	0.71	0.83	0.50
EEG Eye State	Corr.	0.76	0.77	0.76	0.57	0.40
	DTW	0.73	0.87	**0.79**	0.57	0.43
	ED	0.63	0.77	0.69	0.58	0.36
	SAX	0.58	0.80	0.67	0.40	0.37
	SRe	0.59	0.92	0.72	**0.62**	**0.62**
CAN 1 (with heterogeneous horizontal scaling)	Corr.	0.68	0.93	0.79	0.79	0.75
	DTW	0.93	1.00	0.96	**1.00**	**1.00**
	ED	0.68	0.93	0.79	0.90	0.78
	SAX	0.74	0.86	0.80	0.90	0.81
	SRe	0.99	0.99	**0.99**	**1.00**	**1.00**
CAN 2 (with inter-track time shifts)	Corr.	0.82	0.99	0.90	0.97	0.97
	DTW	0.80	0.75	0.77	0.89	0.70
	ED	1.00	0.78	0.87	0.94	0.94
	SAX	0.76	0.93	0.84	0.84	0.84
	SRe	1.00	1.00	**1.00**	**1.00**	**1.00**

Figure 4: ***Accuracy benchmark:*** *best performance of all methods. Complete version in Figure E.7. Best F1-score, mAP30 and mAP50 among five methods are highlighted bold and red. Each method suits different datasets. The proposed method (denoted as SRe) outperforms the other, when there are heterogeneous horizontal scaling or inter-track time shifts, as in the last two cases.*

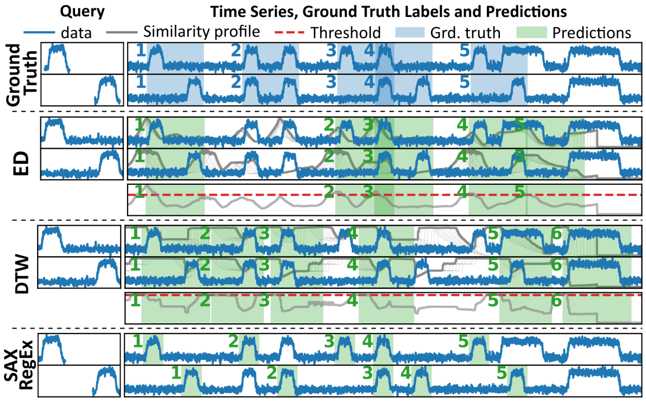

Figure 5: ***Visual inspection:*** *result on the dataset CAN 2. SAXRegEx perfectly finds inter-track time-shifted patterns, while the benchmark methods suffer from false negatives and false positives. Description of the elements in the figure as well as complete version for all methods and datasets are in Appendix D.*

in Figure F.8 in the appendix. In Figure F.9, we report on the relative performance gain compared to DTW.

In short, `SAXRegEx` is nearly x50 faster than DTW and x4.6 faster than MASS, the so-far fastest similarity search tool for time series retrieval. We attribute this speed boost to two reasons. First, `SAXRegEx` naturally captures horizontally scaled patterns. In contrast, the benchmarking methods use eight sliding windows with increasing window lengths, costing roughly x8 the time. However, even reducing the number of sliding windows to four, which is already too coarse, `SAXRegEx` still outperforms all benchmarking methods. Second, `SAXRegEx` has the notion of early termination as soon as the search engine notices a partial mismatch.

5. Discussion and Limitations

Our industry collaborator IAV deploys `SAXRegEx` in a software-as-a-service environment and applies it to data from engine control units, transmission control units and CAN bus data, where heterogeneous horizontal scaling of the pattern and inter-track time shifts occur regularly. It satisfactorily meets automotive engineers' flexible needs for prompt search results.

`SAXRegEx` does not calculate similarity profiles. It makes sense to use a relatively large threshold, tolerating more false positives and filter the predictions with fine-grained distance metrics.

Our current query expansion does not cover vertical translation and scaling of the pattern perpendicular to the time axis. The user should ensure that they do not affect the patterns in the time series. One remedy for the vertical translation problem can be the usage of the derivatives instead of the original data.

We plan to investigate the time complexity of `SAXRegEx` and carry out experiments to study its scalability with increasing query sizes (especially after SAX) and the increasing number of tracks in a contour plot. We would also like to examine the degree of distortions that `SAXRegEx` can bear.

Finally, the current implementation requires a user interface for query definition, progress monitoring, result inspection, and parameter tuning. We plan to collect requirements from the domain experts for the UI, survey the current Visual Query Systems (VQSs), and design a user interface emphasizing the query definition, e.g. for inter-track time-shifted patterns.

6. Conclusion

With `SAXRegEx`, we present a method for multivariate time series retrieval based on SAX, regex, and query expansion. It excels in capturing distorted patterns of various types. In particular, our search engine is ideal to find patterns that are scaled horizontally along the time axis heterogeneously or patterns showing inter-track time shifts, while significantly outperforming state-of-the-art methods in terms of speed. `SAXRegEx` assists automotive engineers quickly find patterns related to various events in the measurement. Nonetheless, the method itself is not limited to any domain-specific prerequisites and can be used in other domains as well. In the future, we plan to design a user interface focusing on the query definition with various kinds of distortions.

References

[AJ02] ANDRÉ-JÖNSSON H.: *Indexing Strategies for Time Series Data*. Department of Computer and Information Science, Linköpings universitet, 2002. 1

[AVG13] A. STEFAN, V. ATHITSOS, G. DAS: The move-split-merge metric for time series. *IEEE Transactions on Knowledge and Data Engineering 25*, 6 (2013), 1425–1438. 2

[BC94] BERNDT D. J., CLIFFORD J.: Using dynamic time warping to find patterns in time series. In *KDD workshop* (1994), vol. 10, pp. 359–370. 2

[BLB*17] BAGNALL A., LINES J., BOSTROM A., LARGE J., KEOGH E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery 31*, 3 (2017), 606–660. 2, 3

[CLL*19] CHENYUN YU, LINTONG LUO, LEANNE LAI-HANG CHAN, THANAWIN RAKTHANMANON, SARANA NUTANONG: A fast lsh-based similarity search method for multivariate time series. *Information Sciences 476* (2019), 337–356. doi:10.1016/j.ins.2018.10.026. 2, 6

[CN04] CHEN L., NG R.: On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30* (2004), pp. 792–803. 2

[GDK*21] GEORGIOS CHATZIGEORGAKIDIS, DIMITRIOS SKOUTAS, KOSTAS PATROUMPAS, THEMIS PALPANAS, SPIROS ATHANASIOU, SPIROS SKIADOPOULOS: Twin subsequence search in time series. *CoRR abs/2104.06874* (2021). 1, 2

[HFZ*15] HUANG Q., FENG J., ZHANG Y., FANG Q., NG W.: Query-aware locality-sensitive hashing for approximate nearest neighbor search. *Proc. VLDB Endow. 9*, 1 (2015), 1–12. doi:10.14778/2850469.2850470. 6

[JDDH19] JOÃO RODRIGUES, DUARTE FOLGADO, DAVID BELO, HUGO GAMBOA: Ssts: A syntactic tool for pattern search on time series. *Information Processing & Management 56*, 1 (2019), 61–76. doi:10.1016/j.ipm.2018.09.001. 2

[JL18a] JIAPING ZHAO, LAURENT ITTI: shapedtw: Shape dynamic time warping. *Pattern Recognition 74* (2018), 171–184. doi:10.1016/j.patcog.2017.09.020. 1, 2

[JL18b] JOSIF GRABOCKA, LARS SCHMIDT-THIEME: Neuralwarp: Time-series similarity with warping networks. *CoRR abs/1812.08306* (2018). 2

[Keo97] KEOGH E.: A fast and robust method for pattern matching in time series databases. *Proceedings of WUSS 97*, 1 (1997), 99. 1, 2

[KP98] KEOGH E. J., PAZZANI M. J.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Kdd* (1998), vol. 98, pp. 239–243. 2

[LKLC03] LIN J., KEOGH E., LONARDI S., CHIU B.: A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (New York, NY, USA, 2003), DMKD '03, Association for Computing Machinery, pp. 2–11. doi:10.1145/882082.882086. 6

[LKWL07] LIN J., KEOGH E., WEI L., LONARDI S.: Experiencing sax: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery 15*, 2 (2007), 107–144. doi:10.1007/s10618-007-0064-z. 2, 3, 6

[LL18] LAFTCHIEV E., LIU Y.: Finding multidimensional patterns in multidimensional time series. In *KDD workshop on MiLeTS 18, London* (2018). 1

[LPH*20] LEKSCHAS F., PETERSON B., HAEHN D., MA E., GEHLENBORG N., PFISTER H.: Peax: Interactive visual pattern search in sequential data using unsupervised deep representation learning. *Computer Graphics Forum 39*, 3 (2020), 167–179. doi:10.1111/cgf.13971. 2

[LS95] LIN R. A. K.-L., SHIM, HARPREET S SAWHNEY KYUSEOK: Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceeding of the 21th International Conference on Very Large Data Bases* (1995), pp. 490–501. 2

[LXZ19] L. HOU, X. JIN, Z. ZHAO: Time series similarity measure via siamese convolutional neural network. In *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)* (2019), pp. 1–6. doi:10.1109/CISP-BMEI48845.2019.8966048. 2

[MA18] MANNINO M., ABOUZIED A.: Qetch: Time series querying with expressive sketches. In *Proceedings of the 2018 International Conference on Management of Data* (New York, NY, USA, 2018), SIGMOD '18, Association for Computing Machinery, pp. 1741–1744. doi:10.1145/3183713.3193547. 2

[MAA*21] MERCER R., ALAEE S., ABDOLI A., SINGH S., MURILLO A., KEOGH E.: Matrix profile xxiii: Contrast profile: A novel time series primitive that allows real world classification. In *ICDM 2021* (2021). 2, 3

[MJE15] MOHAMMAD SHOKOOHI-YEKTA, JUN WANG, EAMONN KEOGH: On the non-trivial generalization of dynamic time warping to the multi-dimensional case. In *Proceedings of the 2015 SIAM International Conference on Data Mining* (2015), pp. 289–297. doi:10.1137/1.9781611974010.33. 1, 2

[MM16] M. CORRELL, M. GLEICHER: The semantics of sketch: Flexibility in visual query systems for time series data. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2016), pp. 131–140. doi:10.1109/VAST.2016.7883519. 2, 3

[PFP21] PENG BOTAO, FATOUROU PANAGIOTA, PALPANAS THEMIS: Sing: Sequence indexing using gpus. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)* (2021), pp. 1883–1888. doi:10.1109/ICDE51399.2021.00171. 2

[SSA*18] S. GHARGHABI, S. IMANI, A. BAGNALL, A. DARVISHZADEH, E. KEOGH: Mpdist: A novel time series distance measure to allow data mining in more challenging scenarios. In *2018 IEEE International Conference on Data Mining (ICDM)* (2018), pp. 965–970. doi:10.1109/ICDM.2018.00119. 1, 2

[YKBB21] YU Y., KRUYFF D., BECKER T., BEHRISCH M.: Pseudo: Interactive pattern search in multivariate time series with locality-sensitive hashing and relevance feedback. *CoRR abs/2104.14962* (2021). URL: https://arxiv.org/abs/2104.14962, arXiv:2104.14962. 2

[YZU*16] YEH CHIN-CHIA MICHAEL, ZHU YAN, ULANOVA LIUDMILA, BEGUM NURJAHAN, DING YIFEI, DAU HOANG ANH, SILVA DIEGO FURTADO, MUEEN ABDULLAH, KEOGH EAMONN: 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)* (2016), pp. 1317–1322. doi:10.1109/ICDM.2016.0179. 2, 3