DeepSketch 2.0: Discovering Temporal Relationships in Large Time Series Datasets

Zheng Zhang
Northwestern University
zheng.zhang@northwestern.edu

Runzhe Jiang
Northwestern University
runzhe.jiang@northwestern.edu

Andrew Crotty
Northwestern University
andrew.crotty@northwestern.edu

***Abstract*—Sketch-based query interfaces have become a popular tool for the interactive exploration of time series datasets because they allow domain experts to search for patterns of interest without writing any code. However, most current systems focus on single-pattern queries, failing to support more complex search behaviors like simultaneously querying for multiple patterns with temporal relationships. This demo will showcase our ongoing work on DEEPSKETCH, a query sketching tool for time series similarity search that addresses these limitations. To support flexible ad hoc queries involving multiple temporally related patterns, DEEPSKETCH provides a search interface that allows users to compose, constrain, and link complex queries. Participants will have the opportunity to interactively explore several large, real-world time series datasets using our tool.**

I. INTRODUCTION

Time series similarity search is a well-studied problem [1]–[3] where the goal is to find the closest matches in a dataset for a given query according to some user-specified distance measure. The advent of sketch-based query interfaces [4]–[16] has empowered domain experts in a multitude of fields to explore datasets without needing programming experience in languages like SQL, Python, or R.

We have built one such tool, called DEEPSKETCH [17], that leverages our *Deep Time Series Similarity Search* (DTS3) pluggable indexing pipeline to scale to much larger time series datasets than current approaches allow. While most existing tools rely on brute-force search, DTS3 trains a foundation model that can produce embeddings for arbitrary user-specified distance measures, enabling fast approximate similarity search via any off-the-shelf vector DBMS.

Like most other sketch-based tools, our initial prototype focused on identifying all occurrences of a single pattern in a dataset, with matches ranked in decreasing order of similarity. However, we found that this design could not support an important class of use cases that involve understanding the temporal relationships *between* multiple different patterns, potentially across more than one time series.

As a concrete example, consider a stock trader who would like to analyze historical market trends in order to develop a data-driven trading strategy. One popular methodology is technical analysis, which involves identifying occurrences of specific patterns in a price action chart that may indicate future price movements. For instance, the line chart in Fig. 1 shows one year of daily closing prices for the S&P 500. The legend ① depicts a common technical indicator called the head-and-shoulders pattern, which consists of one tall central peak (i.e., the “head”) with two shorter peaks of approximately equal height on either side (i.e., the “shoulders”). The shaded blue regions in the figure highlight five instances of the head-and-shoulders pattern, but simply identifying these occurrences has little practical value on its own. As shown, the match in June ② precedes a steep price drop, whereas a positive upward trend actually follows the match in October ③. Knowing that a head-and-shoulders pattern often indicates a bullish-to-bearish trend reversal, the trader can then search for downward price movements that occur only immediately after instances of the first pattern. Shaded in orange, these bearish reversals follow a head-and-shoulders match in four out of five cases, suggesting that the pattern can serve as a reliable predictor.

In this example, the trader had to search for two different patterns with the constraint that one must occur immediately following the other, which is difficult to achieve in existing sketch-based query tools. This demo will highlight our ongoing work to support such use cases in DEEPSKETCH. As shown in Fig. 2, the redesigned web interface allows users to concurrently search for multiple patterns, link those queries to enforce temporal constraints, and overlay match results to visualize the temporal relationships between patterns.

II. BACKGROUND & RELATED WORK

In the following, we first explain time series similarity search and some common distance measures. Then, we discuss related work on visual exploration of time series datasets. Lastly, we describe our pluggable indexing pipeline.

A. Distance Measures

Given a query, the goal in time series similarity search is to locate the top-k most similar matches in a dataset based on a distance measure $d(x_1, x_2)$ that captures the degree to which two time series x_1 and x_2 are alike. Lower distances indicate greater similarity, but the actual distance values need not have any inherent meaning. For instance, if $d(x_1, x_2)$ is half of $d(x_1, x_3)$, then x_1 is more similar to x_2 than x_3, though not necessarily twice as similar. Moreover, the values returned by different distance measures are typically not directly comparable. When choosing a distance measure, the user must carefully consider the specifics of the use case to strike a balance between match accuracy and search speed.

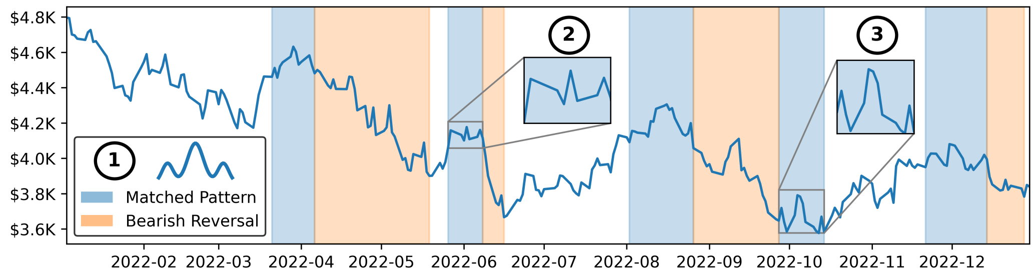

Fig. 1. Line chart of the S&P 500 daily closing prices over one year (Jan–Dec '22). The head-and-shoulders pattern shown in the legend ① is a common indicator used in technical analysis. However, simply finding examples of this pattern (blue regions) offers little benefit without understanding how well it predicts future price movements. In some cases ②, prices fall after a match whereas in others ③, they do not. Identifying price declines that occur only after a head-and-shoulders match (orange regions) can help determine whether the original pattern can accurately predict bullish-to-bearish trend reversals.

The most straightforward distance measures perform simple lock-step comparisons between x_1 and x_2; for example, Euclidean distance (ED) computes the sum of squared pointwise differences. Although cheap to calculate, ED has several shortcomings that make it a poor fit for sketch-based query interfaces, including high sensitivity to noise and local misalignment. Some variants use sliding windows (e.g., shape-based distance [18]) to solve simple issues like shifts or translations, but they cannot handle arbitrary stretching or squeezing of the pattern.

Alternatives like dynamic time warping (DTW) provide a robust solution to these problems by finding an optimal alignment between x_1 and x_2, yielding higher-quality matches but also increasing the computational cost. To accelerate the search, some approaches [19], [20] have combined DTW with ED by leveraging the triangle inequality between the two (i.e., low ED implies low DTW, but the inverse is not necessarily true). Yet, despite their widespread use, traditional distance measures like ED and DTW frequently miss visually similar matches when dealing with hand-drawn queries [5], [11].

More recently, new algorithms have attempted to explicitly capture the most salient visual features when identifying matches. For example, Qetch [11], [12] is based on user studies that showed hand-drawn sketches tend to exaggerate certain aspects (e.g., steepness of slopes, size of peaks/troughs), with the overall shape being more important than minor variations in the pattern. However, designing efficient index structures for complex algorithms like Qetch can be difficult, often forcing them to rely on brute-force search. Other approaches [21], [22] use autoencoders based on a learned embedding space, but they entail either manual data labeling or costly retraining for each new dataset. In summary, all of these algorithms produce subjectively better matches than the traditional alternatives but fundamentally lack scalability.

B. Visual Time Series Exploration

Summary visualizations are a natural way to get a big-picture understanding of time series datasets, including general trends and common patterns. In particular, sketch-based tools [4]–[16] have a relatively low barrier to entry, though many limit queries to a single pattern or allow search over only one time series. On the other hand, interfaces specifically designed for multivariate time series analysis help users to better understand how different time series relate to each other, but they tend to focus primarily on tasks like anomaly detection [23] or forecasting [24]. Even those that explicitly target pattern refinement and search for multivariate datasets (e.g., PSEUDo [25]) only support lock-step pattern matching across series.

C. DTS3: Deep Time Series Similarity Search

As explained, existing approaches either produce poor perceptual matches or cannot scale to large datasets. These problems are only exacerbated when concurrently searching for multiple temporally related patterns across one or more different time series. Our new pluggable indexing pipeline, called *Deep Time Series Similarity Search* (DTS3) [17], seeks to address both of these challenges. Note that DTS3 is *not* a new distance measure—rather, it is a framework for accelerating similarity search for existing distance measures.

The DTS3 pipeline begins by training an autoencoder-based foundation model on a large and diverse corpus of curated time series data to approximate a user-specified distance measure $d(x_1, x_2)$, which can then produce embeddings for any target dataset. Storing these embeddings in a vector DBMS enables fast similarity search that involves using this same foundation model to generate an embedding for the query (e.g., the head-and-shoulders pattern from Fig. 1) and then retrieving the top-k most similar matches for that embedding. Note also that DTS3 can support completely ad hoc queries and does not require the user to specify query patterns a priori.

Previous work in this space (e.g., Peax [21], SEAnet [26], [27], LineNet [22]) leveraged autoencoders to embed time series for similarity search, as well as other use cases like semantic compression [28] and outlier detection [29]. As mentioned, these approaches must usually retrain from scratch on each new dataset, leading to substantial upfront costs and difficulty adapting to change (e.g., evolving datasets, out-of-distribution queries). Instead, DTS3 trains a single foundation model for a distance measure that can be reused across a range of diverse datasets. Although the resulting foundation models can already generalize well, we also provide an optional step for lightweight fine-tuning to further improve match quality for the target dataset.

Fig. 2. DEEPSKETCH web interface for visual exploration and sketch-based querying of time series datasets. Users can draw patterns of interest in the query sketch canvas (bottom right), view results and link queries in the match panel (bottom left), and examine temporal relationships in the result inspector (top).

DTS3 offers two main advantages that allow scaling even complex distance measures to much larger datasets. First, converting time series to an embedding space replaces a potentially expensive distance measure with cheap vector similarity calculations (e.g., ED, cosine similarity). Second, a vector DBMS can now provide efficient similarity search for any distance measure, obviating the need for brute-force search when algorithms lack specialized index structures.

III. DEEPSKETCH 2.0

Our latest version of DEEPSKETCH focuses on supporting richer queries that go beyond simple searches for individual patterns, helping users discover temporal relationships between patterns and across time series. Fig. 2 shows the redesigned DEEPSKETCH web interface, which consists of three main components: (1) the query sketch canvas, where users can draw patterns and specify other query parameters; (2) the match panel, which displays the most similar results for each sketch and allows users to link queries with temporal constraints; and (3) the result inspector, where users can visualize the overlaid matches and explore how patterns relate.

A. Query Sketch Canvas

Users can begin exploring a dataset by drawing a pattern of interest, such as the head-and-shoulders example from Fig. 1, on the query sketch canvas (bottom right). Additional query parameters include: (1) specific time series to search over; (2) the distance measure; (3) the granularity and range of the search window; and (4) a limit on the number of matches to return. DEEPSKETCH can push these constraints down to the vector DBMS in order to filter out data before performing the similarity search. This panel also provides drop-downs for predefined and previously issued queries, presenting further opportunities for caching and reuse.

Fig. 2 shows the query results for the two patterns from Fig. 1, with matches for each highlighted in blue (head-and-shoulders) and red (decline). In contrast to many tools that can identify only fixed-length matches, DEEPSKETCH allows users to specify window size ranges rather than a single value, which is useful for handling uncertainty about the precise time horizon or allowing some fuzziness in the match length. This flexibility is important in real-world settings, where matches are rarely exact; for instance, the patterns in Fig. 1 span anywhere from one to several weeks. Similarly, window granularity selection means that users can concurrently search for patterns that occur on completely different timescales (e.g., short-term price action over a few days leading to a longer-term trend over the following weeks).

B. Match Panel

Query results appear as a ranked list in the preview area (bottom left), with matches indicated by shaded regions. Selecting a series will populate the result inspector (top) with a larger view overlaid on previous query results. Importantly, users can place additional constraints on the matches via the filter drop-down—including linking queries together to discover temporal relationships—that DEEPSKETCH can also push down to the vector DBMS. For example, the price declines in Fig. 1 should occur immediately after the head-and-shoulders matches, such that DEEPSKETCH can substantially narrow the search ranges for the second query.

Above the list of query results, DEEPSKETCH also includes a histogram showing the distribution of match distances. In some cases (e.g., ED), these distances have an intrinsic meaning, but other distance measures (e.g., Qetch) convey only relative similarity, which can confuse users looking to apply threshold-based filtering to matches. The distance distribution histogram addresses this ambiguity by helping users reason about relative match quality. Rather than guessing what constitutes a reasonable match threshold, they can interactively adjust the slider on the histogram to filter matches and observe how the query results in the preview area update, making it much easier to understand the sensitivity of a given distance measure to this parameter.

C. Result Inspector

Finally, users can review the matches, which are overlaid on previous query results and aligned on the time axis to help identify potential temporal relationships between patterns, in the result inspector. The displayed query results can come from either the same time series, as shown in Fig. 2, or multiple different time series in the dataset.

IV. DEMO DESCRIPTION

During the demo, participants can access a hosted version of the DEEPSKETCH web interface using their own devices or one of the provided tablets for a pen-and-touch experience [5], [17], [30]. To introduce them to the tool, we will begin with a guided tutorial based on the example from Fig. 1, after which they can freely explore the available datasets at their leisure. Although our description of DEEPSKETCH has focused primarily on the technical analysis of stock data, we will also include time series datasets from other domains.

ACKNOWLEDGMENTS

This work was supported in part by an award from Google's Research Scholar Program.

REFERENCES

[1] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. J. Keogh, "Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures," *PVLDB*, vol. 1, no. 2, pp. 1542–1552, 2008.

[2] K. Echihabi, K. Zoumpatianos, T. Palpanas, and H. Benbrahim, "The Lernaean Hydra of Data Series Similarity Search: An Experimental Evaluation of the State of the Art," *PVLDB*, vol. 12, no. 2, pp. 112–127, 2018.

[3] ——, "Return of the Lernaean Hydra: Experimental Evaluation of Data Series Approximate Similarity Search," *PVLDB*, vol. 13, no. 3, pp. 403–420, 2019.

[4] M. Wattenberg, "Sketching a Graph to Query a Time-Series Database," in *CHI EA*, 2001, pp. 381–382.

[5] P. Eichmann and E. Zgraggen, "Evaluating Subjective Accuracy in Time Series Pattern-Matching Using Human-Annotated Rankings," in *IUI*, 2015, pp. 28–37.

[6] P. K. Muthumanickam, K. Vrotsou, M. Cooper, and J. Johansson, "Shape Grammar Extraction for Efficient Query-by-Sketch Pattern Matching in Long Time Series," in *VAST*, 2016, pp. 121–130.

[7] M. Correll and M. Gleicher, "The Semantics of Sketch: Flexibility In Visual Query Systems For Time Series Data," in *VAST*, 2016, pp. 131–140.

[8] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, and A. G. Parameswaran, "Effortless Data Exploration with zenvisage: An Expressive and Interactive Visual Analytics System," *PVLDB*, vol. 10, no. 4, pp. 457–468, 2016.

[9] T. Siddiqui, J. Lee, A. Kim, E. Xue, X. Yu, S. Zou, L. Guo, C. Liu, C. Wang, K. Karahalios, and A. G. Parameswaran, "Fast-Forwarding to Desired Visualizations with zenvisage," in *CIDR*, 2017.

[10] D. J. L. Lee, J. Lee, T. Siddiqui, J. Kim, K. Karahalios, and A. G. Parameswaran, "You can't always sketch what you want: Understanding Sensemaking in Visual Query Systems," *TVCG*, vol. 26, no. 1, pp. 1267–1277, 2020.

[11] M. Mannino and A. Abouzied, "Expressive Time Series Querying with Hand-Drawn Scale-Free Sketches," in *CHI*, 2018, p. 388.

[12] ——, "Qetch: Time Series Querying with Expressive Sketches," in *SIGMOD*, 2018, pp. 1741–1744.

[13] T. Siddiqui, P. Luh, Z. Wang, K. Karahalios, and A. G. Parameswaran, "ShapeSearch: Flexible Pattern-based Querying of Trend Line Visualizations," *PVLDB*, vol. 11, no. 12, pp. 1962–1965, 2018.

[14] ——, "ShapeSearch: A Flexible and Efficient System for Shape-based Exploration of Trendlines," in *SIGMOD*, 2020, pp. 51–65.

[15] C. Fan, K. Matkovic, and H. Hauser, "Sketch-Based Fast and Accurate Querying of Time Series Using Parameter-Sharing LSTM Networks," *TVCG*, vol. 27, no. 12, pp. 4495–4506, 2021.

[16] L. Yan, N. Xu, G. Li, S. S. Bhowmick, B. Choi, and J. Xu, "SENSOR: Data-driven Construction of Sketch-based Visual Query Interfaces for Time Series Data," *PVLDB*, vol. 15, no. 12, pp. 3650–3653, 2022.

[17] Z. Zhang, Z. Shao, and A. Crotty, "DeepSketch: A Query Sketching Interface for Deep Time Series Similarity Search," *PVLDB*, vol. 17, no. 12, pp. 4369–4372, 2024.

[18] J. Paparrizos and L. Gravano, "k-Shape: Efficient and Accurate Clustering of Time Series," in *SIGMOD*, 2015, pp. 1855–1870.

[19] R. Neamtu, R. Ahsan, E. A. Rundensteiner, and G. N. Sárközy, "Interactive Time Series Exploration Powered by the Marriage of Similarity Distances," *PVLDB*, vol. 10, no. 3, pp. 169–180, 2016.

[20] R. Neamtu, R. Ahsan, C. Lovering, C. Nguyen, E. A. Rundensteiner, and G. N. Sárközy, "Interactive Time Series Analytics Powered by ONEX," in *SIGMOD*, 2017, pp. 1595–1598.

[21] F. Lekschas, B. Peterson, D. Haehn, E. Ma, N. Gehlenborg, and H. Pfister, "Peax: Interactive Visual Pattern Search in Sequential Data Using Unsupervised Deep Representation Learning," *CGF*, vol. 39, no. 3, pp. 167–179, 2020.

[22] Y. Luo, Y. Zhou, N. Tang, G. Li, C. Chai, and L. Shen, "Learned Data-aware Image Representations of Line Charts for Similarity Search," *PACMMOD*, vol. 1, no. 1, pp. 88:1–88:29, 2023.

[23] D. Liu, S. Alnegheimish, A. Zytek, and K. Veeramachaneni, "MTV: Visual Analytics for Detecting, Investigating, and Annotating Anomalies in Multivariate Time Series," *PACMHCI*, vol. 6, no. CSCW1, pp. 103:1–103:30, 2022.

[24] K. Xu, J. Yuan, Y. Wang, C. T. Silva, and E. Bertini, "mTSeer: Interactive Visual Exploration of Models on Multivariate Time-series Forecast," in *CHI*, 2021, pp. 23:1–23:15.

[25] Y. Yu, D. Kruyff, J. Jiao, T. Becker, and M. Behrisch, "PSEUDo: Interactive Pattern Search in Multivariate Time Series with Locality-Sensitive Hashing and Relevance Feedback," *TVCG*, vol. 29, no. 1, pp. 33–42, 2023.

[26] Q. Wang and T. Palpanas, "Deep Learning Embeddings for Data Series Similarity Search," in *KDD*, 2021, pp. 1708–1716.

[27] ——, "SEAnet: A Deep Learning Architecture for Data Series Similarity Search," *TKDE*, vol. 35, no. 12, pp. 12 972–12 986, 2023.

[28] A. Ilkhechi, A. Crotty, A. Galakatos, Y. Mao, G. Fan, X. Shi, and U. Çetintemel, "DeepSqueeze: Deep Semantic Compression for Tabular Data," in *SIGMOD*, 2020, pp. 1733–1746.

[29] D. Campos, T. Kieu, C. Guo, F. Huang, K. Zheng, B. Yang, and C. S. Jensen, "Unsupervised Time Series Outlier Detection with Diversity-Driven Convolutional Ensembles," *PVLDB*, vol. 15, no. 3, pp. 611–623, 2021.

[30] A. Crotty, A. Galakatos, E. Zgraggen, C. Binnig, and T. Kraska, "Vizdom: Interactive Analytics through Pen and Touch," *PVLDB*, vol. 8, no. 12, pp. 2024–2027, 2015.