DeepSketch: A Query Sketching Interface for Deep Time Series Similarity Search

Zheng Zhang
Northwestern University
zheng.zhang@northwestern.edu

Zhuhan Shao
Northwestern University
zhuhan.shao@northwestern.edu

Andrew Crotty
Northwestern University
andrew.crotty@northwestern.edu

ABSTRACT

By empowering domain experts to perform interactive exploration of large time series datasets, sketch-based query interfaces have revitalized interest in the well-studied problem of time series similarity search. In this new interaction paradigm, recent similarity algorithms (e.g., Qetch, Peax, LineNet) that attempt to capture perceptually relevant features have supplanted older, more straightforward distance measures (e.g., Euclidean, DTW). However, the downside of these algorithms is the resulting difficulty in designing corresponding index structures to support efficient similarity search over large datasets, thus necessitating brute-force search.

This demo will showcase *Deep Time Series Similarity Search* (DTS3), our pluggable indexing pipeline for arbitrary distance measures. DTS3 can automatically train a foundation model for any custom, user-supplied distance measure with no strict constraints (e.g., differentiability), thus enabling fast retrieval via an off-the-shelf vector DBMS. Using our DeepSketch web interface, participants can compare DTS3 to the baseline brute-force versions of several similarity algorithms to see that our approach can achieve much lower latency without sacrificing accuracy when searching over large, real-world time series datasets.

PVLDB Reference Format:
Zheng Zhang, Zhuhan Shao, and Andrew Crotty. DeepSketch: A Query Sketching Interface for Deep Time Series Similarity Search. PVLDB, 17(12): 4369 - 4372, 2024.
doi:10.14778/3685800.3685877

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/andrewcrotty/dts3.

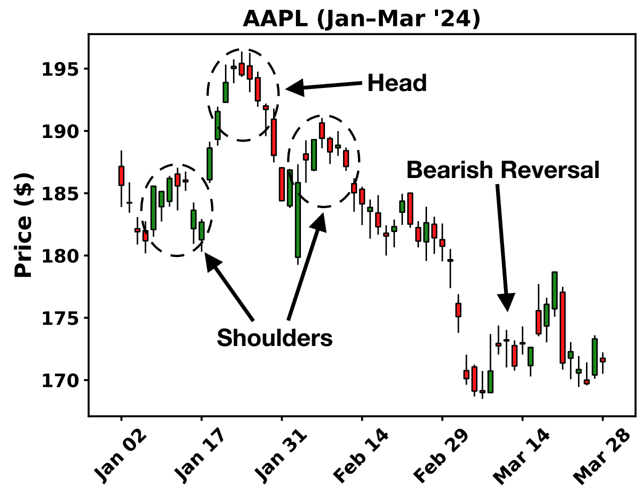

Figure 1: Daily candlestick chart for Apple stock (AAPL) from Jan–Mar '24. The pictured head-and-shoulders technical pattern can indicate a bullish-to-bearish trend reversal.

1 INTRODUCTION

Time series similarity search is an extremely well-studied problem [4, 6, 7], with applications in a wide variety of domains from finance to medicine. Sketch-based query interfaces [2, 8, 9, 11, 14–16, 19–22, 25, 26] enable domain experts in these areas to interactively query large time series datasets simply by drawing patterns of interest, without requiring them to have any programming experience. As a concrete example, consider a stock analyst who would like to search for technical patterns in a dataset containing historical daily stock prices. The goal of technical analysis is to identify patterns in a stock chart that potentially predict future price movements, such as the head-and-shoulders pattern shown in Figure 1. After searching the dataset to find matches to the hand-drawn query, the analyst can then review the results to formulate a data-driven trading strategy.

Many existing approaches to time series similarity search rely on widely used distance measures such as the relatively cheap Euclidean distance (ED), the more expensive but better dynamic time warping (DTW), or even combinations of the two [17, 18] to reduce search cost while still providing good matches. Unfortunately, these distance measures often perform poorly in sketch-based interfaces due to difficulty with common characteristics of hand-drawn queries (e.g., noise, local distortions) [14]. Several recent algorithms (e.g., Qetch [14, 15], Peax [12], LineNet [13]) therefore attempt to more accurately model how humans perceive visual matches.

However, unlike well-established distance measures like ED, these newer algorithms usually lack index structures necessary for efficient match retrieval, meaning they must resort to brute-force search. Even approaches based on autoencoders, which generate time series embeddings that a vector DBMS could index, require training from scratch for each new dataset, sometimes with manual data curation or labeling. Moreover, they typically do not generalize well to drift in streaming data or queries that deviate too far from the training set, which would again require expensive retraining in order to adapt. As such, these algorithms generally cannot scale to large datasets.

Proceedings of the VLDB Endowment, Vol. 17, No. 12 ISSN 2150-8097.
doi:10.14778/3685800.3685877

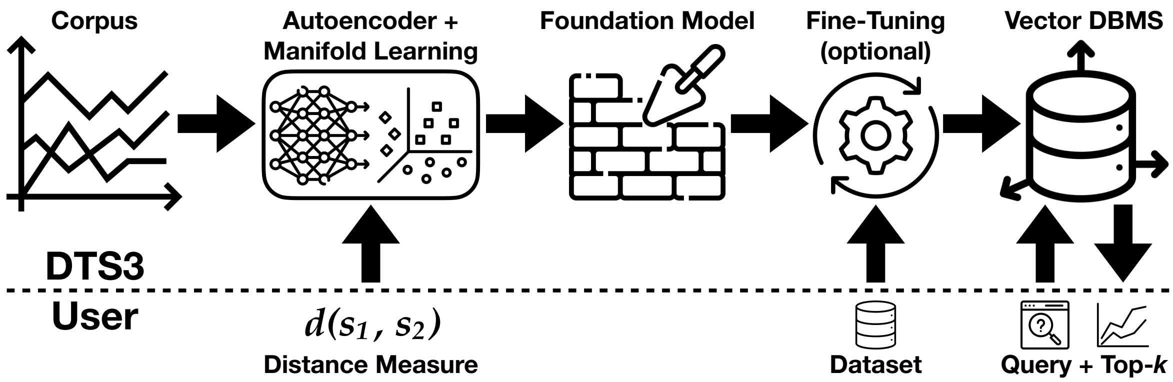

Figure 2: Overview of the *Deep Time Series Similarity Search* (DTS3) pluggable indexing pipeline. Given an arbitrary user-supplied distance measure, DTS3 automatically trains a foundation model capable of generating embeddings for storage in an off-the-shelf vector DBMS. After optional fine-tuning, the user can query the indexed dataset stored in the vector DBMS.

To solve these problems, we propose a new pluggable indexing pipeline called *Deep Time Series Similarity Search* (DTS3) that permits efficient match retrieval without brute-force search. When a user provides a custom distance measure, our DTS3 pipeline automatically trains a foundation model capable of generating meaningful embeddings for arbitrary time series datasets. Importantly, DTS3 imposes no strict requirements on the distance measure (e.g., differentiability), and the resulting embeddings can be indexed by an off-the-shelf vector DBMS.

This demo will allow participants to evaluate DTS3 using our DeepSketch web interface. After a brief tutorial, they will have the opportunity to freely explore different time series datasets to see how DTS3 performs relative to brute-force search for various similarity algorithms in terms of both speed and match quality.

2 DTS3

The goal of time series similarity search is to find the top-k most similar matches for a given query. As a proxy for the similarity between two time series s_1 and s_2, the user selects a distance measure $d(s_1, s_2)$ where lower distance values indicate higher similarity, though the values returned by different distance measures are usually 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.

The simplest distance measures perform pointwise calculations on the two time series. For example, Euclidean distance (ED) computes the sum of the differences between s_1 and s_2. Although straightforward and 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.

Better distance measures like dynamic time warping (DTW) address the misalignment issue by first finding an optimal alignment between s_1 and s_2, yielding higher-quality matches but also increasing the computational cost. To solve this problem, some approaches [17, 18] have combined DTW with ED to accelerate the distance calculation by leveraging the triangle inequality between the two (i.e., a low ED distance implies a low DTW distance, but the inverse is not necessarily true). Yet, despite their widespread use, traditional distance measures like ED and DTW were not designed for hand-drawn queries and frequently miss visually similar matches [8, 14].

Therefore, new similarity algorithms that explicitly target the most salient visual features when identifying matches have emerged in recent years. For example, Qetch [14, 15] is based on user study results showing that 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. Due to their complexity, though, designing efficient index structures for algorithms like Qetch can prove difficult, which forces them to rely on brute-force search. Other approaches [12, 13] in this category use autoencoders to generate embeddings that, incidentally, are amenable to storage in a vector DBMS, but they entail either manual data labeling or costly training for each new dataset. In summary, all of these algorithms produce subjectively better matches than the traditional alternatives but fundamentally lack scalability.

Our approach, called *Deep Time Series Similarity Search* (DTS3), seeks to overcome all of these challenges. DTS3 is not a new time series similarity algorithm; rather, it serves as a pluggable indexing pipeline for arbitrary distance measures. Figure 2 shows a high-level overview of DTS3, which can be broken down into three main parts: (1) training; (2) indexing; and (3) querying.

2.1 Training

The DTS3 pipeline begins by training an autoencoder to approximate the user-supplied distance measure $d(s_1, s_2)$. Prior work (e.g., Peax [12], SEAnet [23, 24], LineNet [13]) has also used autoencoders to embed time series for similarity search, as well as for other time series use cases (e.g., semantic compression [10], outlier detection [1]). Specifically, we use a topological autoencoder with a manifold learning module as a regularizer [5] to support custom similarity algorithms without strict constraints (e.g., a differentiable loss function). We feed each training batch separately to the autoencoder, which attempts to minimize the reconstruction error, and the manifold learning module, which attempts to preserve the topological structure of the dataset relative to the distance measure in the embedding space. The loss function combines these two terms using a weighted MSE, allowing us to leverage the generalizability of autoencoders while also retaining the most important features for the distance measure in the embeddings.

As mentioned, existing approaches that use autoencoders typically train on the data to be indexed, leading to substantial upfront costs for every new dataset. This requirement presents even more issues for evolving datasets and out-of-distribution queries. Instead, we propose to train a single foundation model for a distance measure that the user can reuse across a range of diverse datasets. DTS3 utilizes a large corpus of real-world time series data that we compiled from an assortment of publicly available datasets and further augmented (e.g., adding noise, random subsampling). Although the resulting foundation models can already generalize well to arbitrary datasets, we also provide an optional step for lightweight fine-tuning to improve accuracy on the target dataset.

2.2 Indexing

In terms of query execution, the benefits of DTS3 are twofold. First, converting time series to an embedding space replaces a potentially expensive distance measure with a much cheaper vector similarity calculation, such as ED or cosine similarity. Second, embeddings unlock the ability for any arbitrary distance measure to use an off-the-shelf vector DBMS for indexing, thereby avoiding brute-force search. Together, these two advantages can enable complex similarity algorithms to scale to much larger datasets.

At the same time, indexing embeddings with a vector DBMS incurs additional storage overhead compared to brute-force search, and the extra space consumption can quickly become significant depending on embedding dimensionality and dataset size. For example, embeddings with the same size as the original time series would effectively double the space consumption, but they would also afford better retrieval accuracy than a lower-dimensional embedding. The user can directly influence this trade-off by specifying the desired embedding size when creating the foundation model. Otherwise, DTS3 will choose sensible defaults in a best-effort attempt to balance accuracy and space overhead.

2.3 Querying

To search for matches in the indexed dataset, the query (e.g., a hand-drawn sketch) is first fed to the trained foundation model to generate a corresponding embedding. Then, DTS3 retrieves the top-k most similar matches to the query embedding from the vector DBMS and returns them to the user. Note that DTS3 does not require the user to specify query patterns a priori and can support completely ad hoc querying.

3 DEEPSKETCH

Figure 3 shows the DeepSketch web interface, which we adapted from Qetch [14, 15]. The tool consists of three main components: (1) dataset browser; (2) sketch area; and (3) result panels.

3.1 Dataset Browser

After selecting an available dataset from the drop-down menu in the top right, the user can immediately begin exploring it via the dataset browser (top). This view is useful for forming a preliminary understanding of the data and generating ideas for potential queries. For example, a user interested in Apple stock (AAPL) can select it from the search bar to get a larger, more detailed view, with time displayed across the x-axis and price on the y-axis. The user can also zoom in on specific time ranges, as well as overlay multiple different time series to compare them side-by-side.

3.2 Sketch Area

The user can issue similarity search queries from the sketch area (bottom right) by drawing a pattern of interest, such as the head-and-shoulders example from Figure 1. The tool also provides a set of predefined queries representing common patterns, as well as a history of previously issued queries that the user may wish to revisit. The user can additionally specify the following optional parameters to further refine the query: (1) specific time series to search; (2) time window granularity; (3) the distance measure to use; and (4) a limit on the number of returned matches. In Figure 3, the user is searching for a head-and-shoulders pattern in AAPL closing prices at a monthly granularity using DTW as the similarity measure, with a limit of 10 results.

3.3 Result Panels

Once the user has submitted a query, matches will begin to populate the result panels (bottom left) in a streaming fashion as they arrive. The panel on the left displays results from the brute-force version of the chosen distance measure, and the panel on the right shows matches retrieved from the vector DBMS using the DTS3 approach. Selecting a match from one of the result panels will pull up a larger, zoomable view with the match overlaid in the dataset browser. The user can sort results by either computed distance or match length, and the adjacent result panels facilitate a clear side-by-side comparison of DTS3 with the brute-force approach. Lastly, each result panel contains a wall clock timer in the top right so the user can evaluate total query execution time.

4 DEMO EXPERIENCE

For the demo, participants can connect to the DeepSketch web interface using either their own device or one of the provided tablets for a pen-and-touch experience [3, 8]. To get a feel for the tool, we will begin by walking them through the stock analysis scenario described in this proposal. Then, we will encourage them to explore the available datasets at their leisure by interactively issuing queries, examining the results, and comparing DTS3 to the brute-force versions of the different similarity algorithms. Although our description of DeepSketch has focused primarily on the technical analysis of stock data, we will also include time series datasets from a variety of other domains.

ACKNOWLEDGMENTS

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

REFERENCES

[1] David Campos, Tung Kieu, Chenjuan Guo, Feiteng Huang, Kai Zheng, Bin Yang, and Christian S. Jensen. 2021. Unsupervised Time Series Outlier Detection with Diversity-Driven Convolutional Ensembles. *PVLDB* 15, 3 (2021), 611–623.

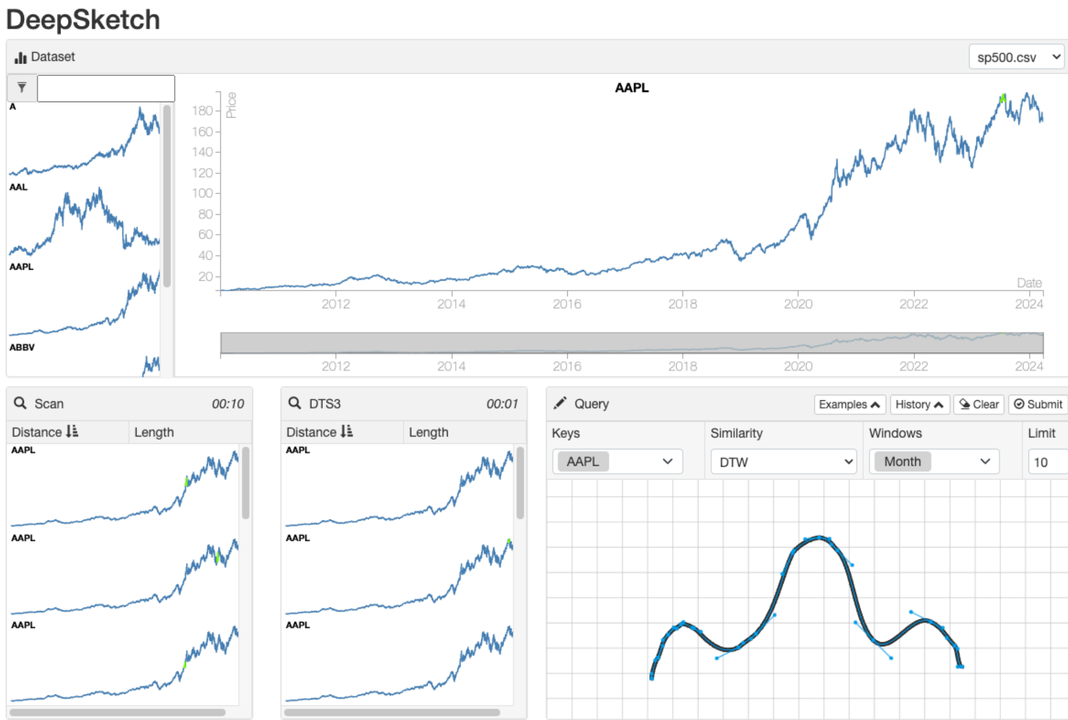

Figure 3: Screenshot of the DeepSketch web interface for visual exploration and sketch-based querying of time series datasets. Participants can get a feel for the data through the dataset browser (top) before issuing hand-drawn queries using the sketch area (bottom right). The result panels (bottom left) will allow participants to directly compare our DTS3 approach with the brute-force version of each distance measure, both in terms of match quality and query execution speed.

[2] Michael Correll and Michael Gleicher. 2016. The Semantics of Sketch: Flexibility In Visual Query Systems For Time Series Data. In *VAST*. 131–140.

[3] Andrew Crotty, Alex Galakatos, Emanuel Zgraggen, Carsten Binnig, and Tim Kraska. 2015. Vizdom: Interactive Analytics through Pen and Touch. *PVLDB* 8, 12 (2015), 2024–2027.

[4] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn J. Keogh. 2008. Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures. *PVLDB* 1, 2 (2008), 1542–1552.

[5] Andrés F. Duque, Sacha Morin, Guy Wolf, and Kevin R. Moon. 2023. Geometry Regularized Autoencoders. *PAMI* 45, 6 (2023), 7381–7394.

[6] 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* 12, 2 (2018), 112–127.

[7] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2019. Return of the Lernaean Hydra: Experimental Evaluation of Data Series Approximate Similarity Search. *PVLDB* 13, 3 (2019), 403–420.

[8] Philipp Eichmann and Emanuel Zgraggen. 2015. Evaluating Subjective Accuracy in Time Series Pattern-Matching Using Human-Annotated Rankings. In *IUI*. 28–37.

[9] Chaoran Fan, Kresimir Matkovic, and Helwig Hauser. 2021. Sketch-Based Fast and Accurate Querying of Time Series Using Parameter-Sharing LSTM Networks. *TVCG* 27, 12 (2021), 4495–4506.

[10] Amir Ilkhechi, Andrew Crotty, Alex Galakatos, Yicong Mao, Grace Fan, Xiran Shi, and Ugur Çetintemel. 2020. DeepSqueeze: Deep Semantic Compression for Tabular Data. In *SIGMOD*. 1733–1746.

[11] Doris Jung Lin Lee, John Lee, Tarique Siddiqui, Jaewoo Kim, Karrie Karahalios, and Aditya G. Parameswaran. 2020. You can't always sketch what you want: Understanding Sensemaking in Visual Query Systems. *TVCG* 26, 1 (2020), 1267–1277.

[12] Fritz Lekschas, Brant Peterson, Daniel Haehn, Eric Ma, Nils Gehlenborg, and Hanspeter Pfister. 2020. Peax: Interactive Visual Pattern Search in Sequential Data Using Unsupervised Deep Representation Learning. *CGF* 39, 3 (2020), 167–179.

[13] Yuyu Luo, Yihui Zhou, Nan Tang, Guoliang Li, Chengliang Chai, and Leixian Shen. 2023. Learned Data-aware Image Representations of Line Charts for Similarity Search. *PACMMOD* 1, 1 (2023), 88:1–88:29.

[14] Miro Mannino and Azza Abouzied. 2018. Expressive Time Series Querying with Hand-Drawn Scale-Free Sketches. In *CHI*. 388.

[15] Miro Mannino and Azza Abouzied. 2018. Qetch: Time Series Querying with Expressive Sketches. In *SIGMOD*. 1741–1744.

[16] Prithiviraj K. Muthumanickam, Katerina Vrotsou, Matthew Cooper, and Jimmy Johansson. 2016. Shape Grammar Extraction for Efficient Query-by-Sketch Pattern Matching in Long Time Series. In *VAST*. 121–130.

[17] Rodica Neamtu, Ramoza Ahsan, Charles Lovering, Cuong Nguyen, Elke A. Rundensteiner, and Gábor N. Sárközy. 2017. Interactive Time Series Analytics Powered by ONEX. In *SIGMOD*. 1595–1598.

[18] Rodica Neamtu, Ramoza Ahsan, Elke A. Rundensteiner, and Gábor N. Sárközy. 2016. Interactive Time Series Exploration Powered by the Marriage of Similarity Distances. *PVLDB* 10, 3 (2016), 169–180.

[19] Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, and Aditya G. Parameswaran. 2016. Effortless Data Exploration with zenvisage: An Expressive and Interactive Visual Analytics System. *PVLDB* 10, 4 (2016), 457–468.

[20] Tarique Siddiqui, John Lee, Albert Kim, Edward Xue, Xiaofo Yu, Sean Zou, Lijin Guo, Changfeng Liu, Chaoran Wang, Karrie Karahalios, and Aditya G. Parameswaran. 2017. Fast-Forwarding to Desired Visualizations with Zenvisage. In *CIDR*.

[21] Tarique Siddiqui, Paul Luh, Zesheng Wang, Karrie Karahalios, and Aditya G. Parameswaran. 2018. ShapeSearch: Flexible Pattern-based Querying of Trend Line Visualizations. *PVLDB* 11, 12 (2018), 1962–1965.

[22] Tarique Siddiqui, Paul Luh, Zesheng Wang, Karrie Karahalios, and Aditya G. Parameswaran. 2020. ShapeSearch: A Flexible and Efficient System for Shape-based Exploration of Trendlines. In *SIGMOD*. 51–65.

[23] Qitong Wang and Themis Palpanas. 2021. Deep Learning Embeddings for Data Series Similarity Search. In *KDD*. 1708–1716.

[24] Qitong Wang and Themis Palpanas. 2023. SEAnet: A Deep Learning Architecture for Data Series Similarity Search. *TKDE* 35, 12 (2023), 12972–12986.

[25] Martin Wattenberg. 2001. Sketching a Graph to Query a Time-Series Database. In *CHI EA*. 381–382.

[26] Li Yan, Nerissa Xu, Guozhong Li, Sourav S. Bhowmick, Byron Choi, and Jianliang Xu. 2022. SENSOR: Data-driven Construction of Sketch-based Visual Query Interfaces for Time Series Data. *PVLDB* 15, 12 (2022), 3650–3653.