



Temporal Similarity and Vector Search: Advanced Techniques for Real-Time Insights



Michael Ryaboy, KX Developer Advocate

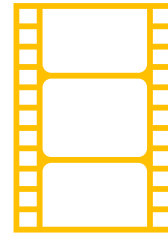
Rise of Unstructured Data



Text



Audio



Video



Image



80% of all data

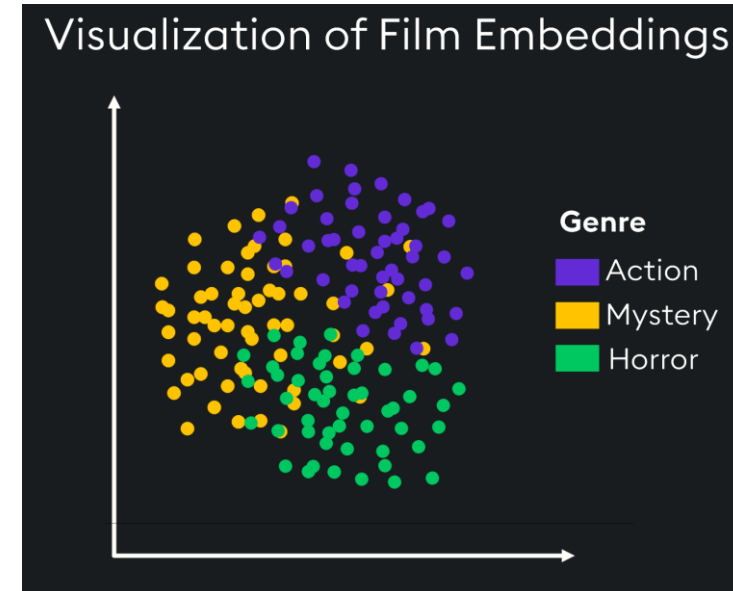
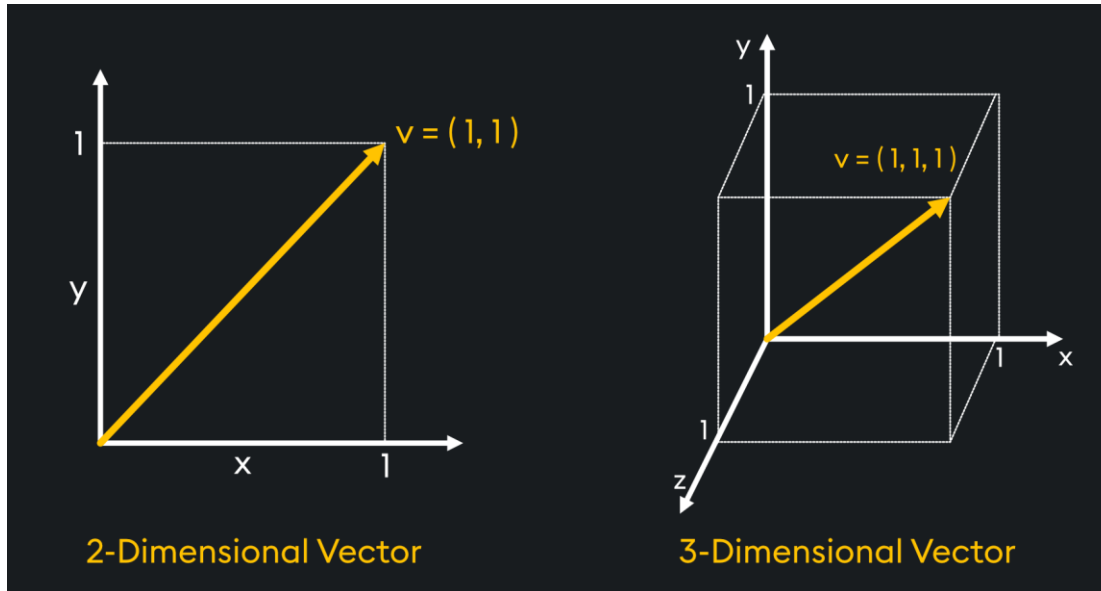
Problem:

Relational Search Methods Fail for Unstructured Data

- Tagging unstructured data accurately is challenging and often incomplete
- Difficult to efficiently find and compare similar items, like images or text
- Generating meaningful tags from user queries is complex and error-prone
- Many similar items may match a search, but there's often no effective way to rank them

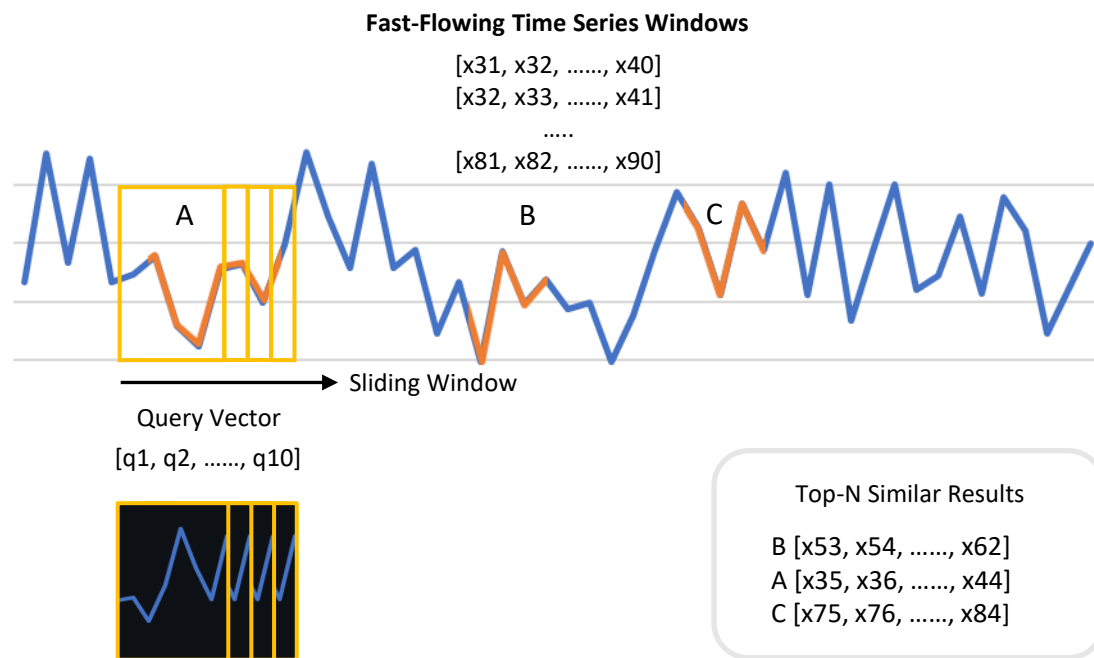
Solution: Represent Data as Vectors

- Machine learning models map data to a vector space
- Similar items mapped close together
- Enables numerical comparisons between items



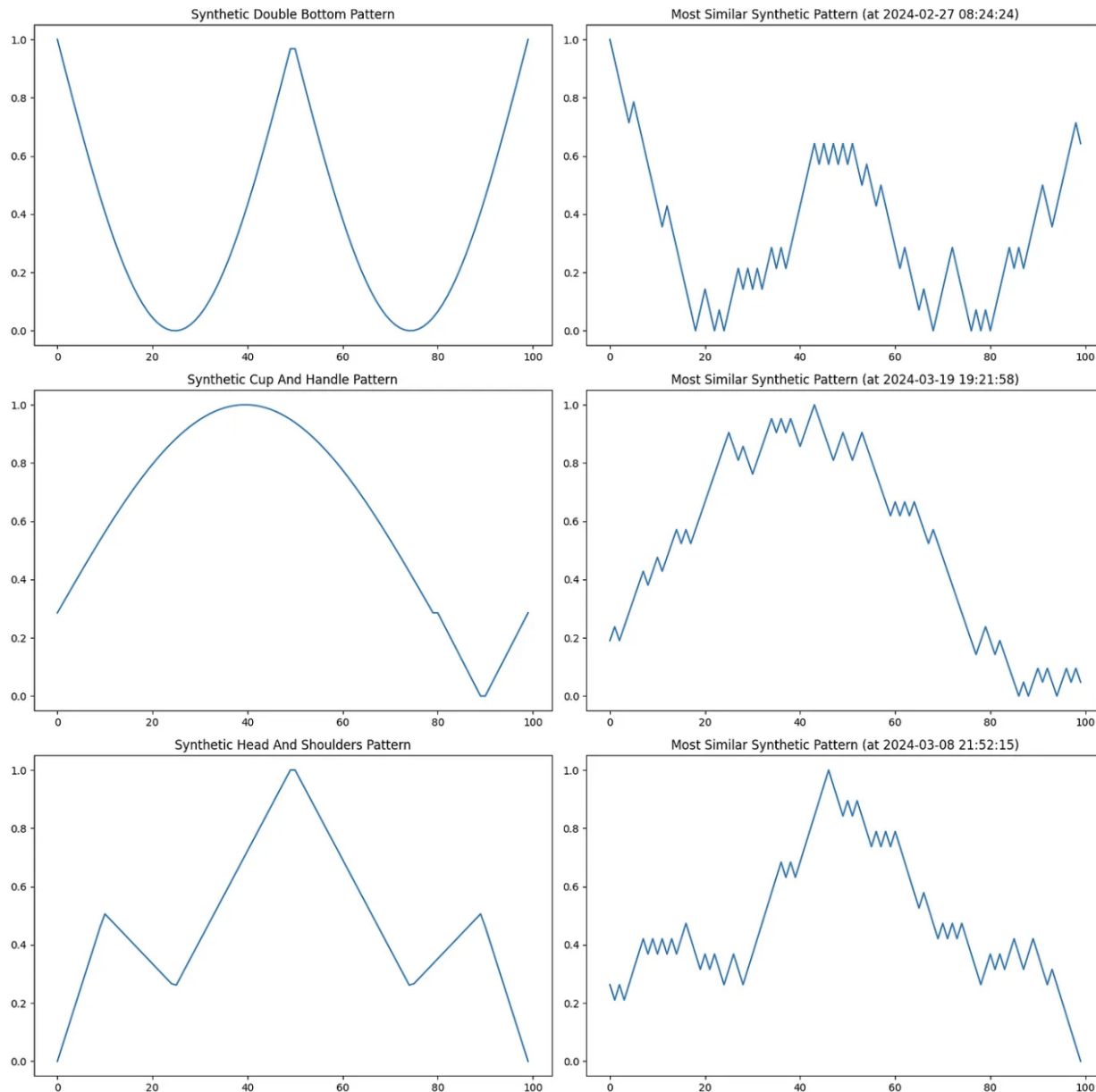
Temporal Similarity Search

- Conduct near real-time similarity search on across high-velocity, streaming data
- Performs vector search directly on time series columns of existing kdb+ data
- No need to embed, extract, or store the vectors in an index
- Performant search with baseline results per search returned in ~8 seconds for searches across ~250M datapoints (run on machine with 8 vCPU 32 GB RAM)
- Dynamic searches: change number of points to match at runtime
- Outlier search to identify anomalies



Find Relationships Between Structured/Unstructured Data

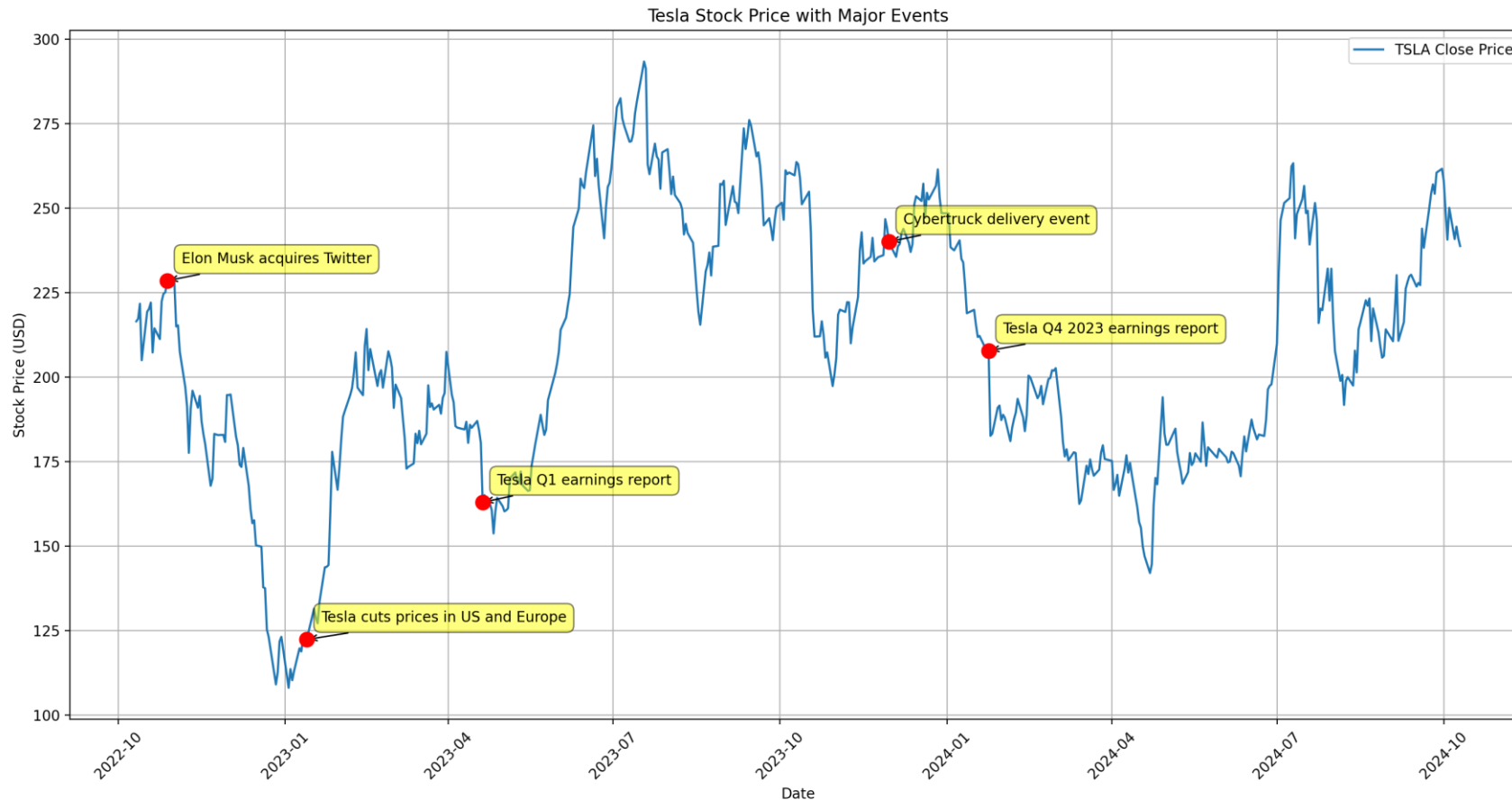
We can also search real and synthetic patterns, and see how the relationships between these patterns and unstructured data



Technical Analysis Patterns

Problem: Real Data Is Unstructured + Structured

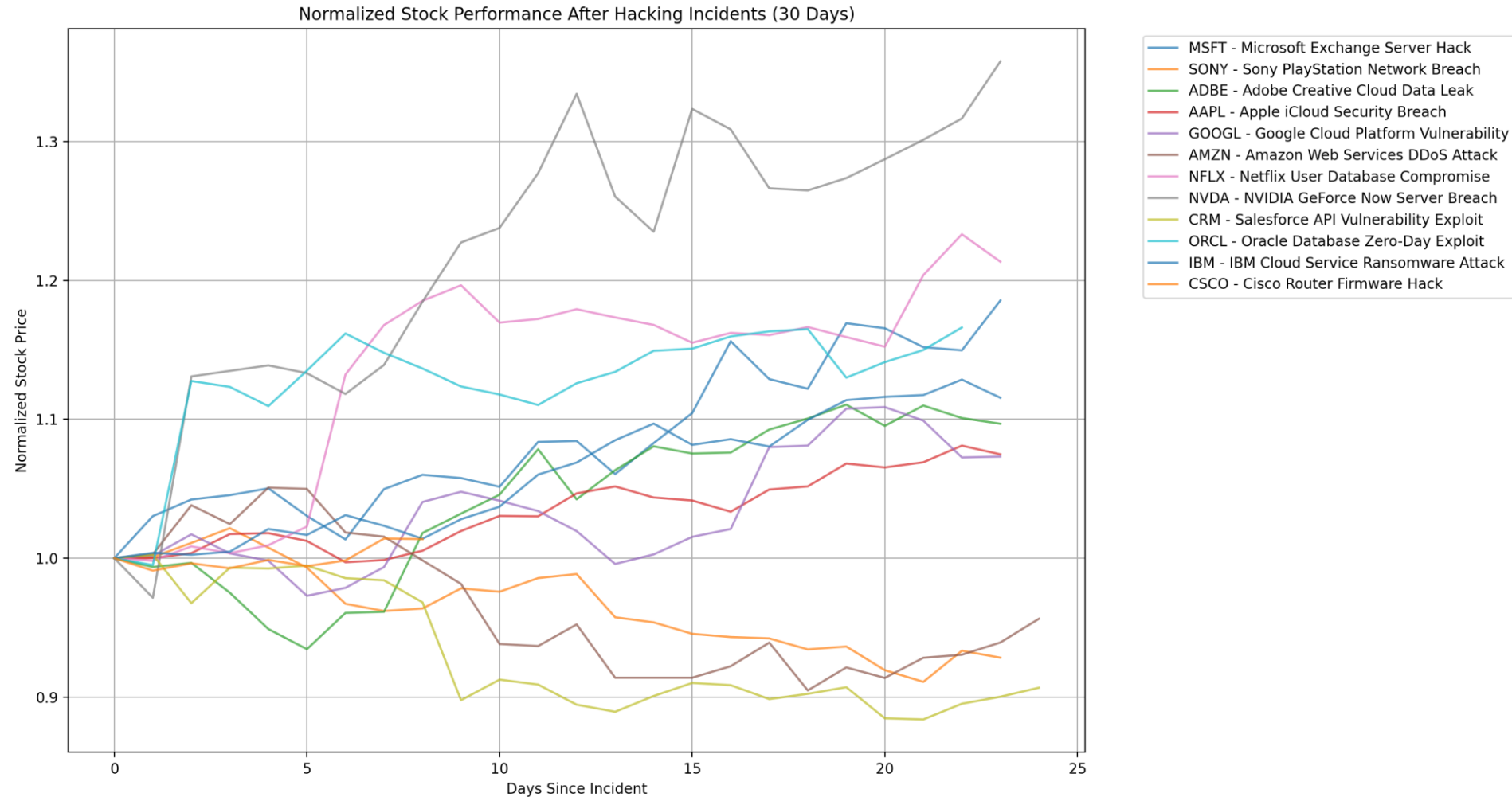
Most of the time, our data isn't just structured, it also includes unstructured data like news events, social media events, and other real-world events. Because this data isn't structured, it's easy to ignore.



Solution: Represent Unstructured Data as Vectors

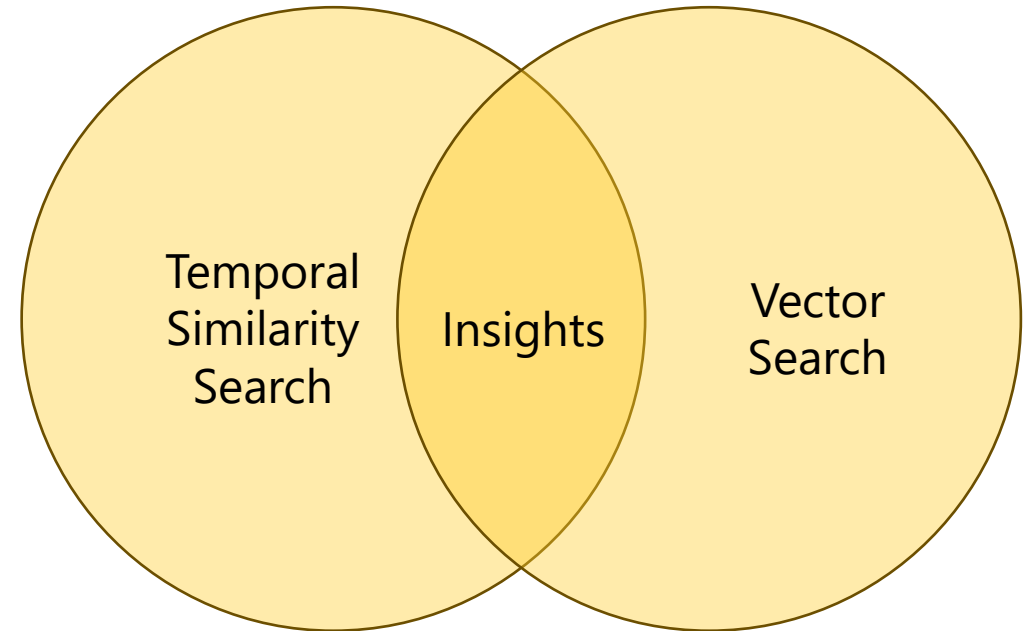
*We can now wait until an event comes in, perform **vector search** for similar events, and then see how our structured data looks. Based on this, we can make predictions / educated decisions.*

Example query: Hacking incident on a major tech company:



Combining Temporal and Vector Search

- Temporal Similarity Search = time-series patterns
- Vector Search = similarities in unstructured data (e.g., news).
- Combined = better insights, prediction
- Focus: search, anomaly detection, prediction



Anomaly Detection Use Cases

Temporal Similarity Search provides the tools to identify patterns, trends, and anomalies in time series datasets by returning the n nearest neighbors for a given pattern/curve. This method compares the **overall shape of curves** rather than relying on more **rigid rule-based approaches**.

- Immediately identify anomalies to **proactively address data feed issues**
- Detect and **address order performance issues** when an unexpected deviation occurs
- Quickly act on anomalies in trading patterns related to **algo execution, toxic order flow, or market abuse**
- Detect changes in customer trading patterns to expedite **proactive customer outreach**



KDB.AI

THANK YOU

Michael Ryaboy
AI Developer Advocate
mryaboy@kx.com