CS E4780 Scalable Systems and Data Management Course Project
Detecting Trading Trends in Financial Tick Data

Responsible teacher: Prof. Bo Zhao
Staff: Dr. Tuo Shi, Zhongxue Xie, Jiaxin Guo, Cong Yu, Linus Jern, Songlin Jiang, Mustapha Abdullahi
cs-e4780@aalto.fi

1 Background

Trading is fuelled by precise real-time event data together with reliable background information about financial instruments such as equities, indices or funds. Instances of a financial instrument are called *symbols*. The high-volume streams of events reporting demand (*ask*), supply (*bid*), made trades (*last*), and other information about each symbol are called financial *market data*.

The overall amount of market data published by the various exchanges on a daily basis and processed by technical solution providers such as Infront Financial Technology GmbH (formerly vwd Vereinigte Wirtschaftsdienste GmbH) is massively increasing. For example, the daily average number of events being processed by Infront increased from 18 billion in 2019 to 40 billion in 2021.

Leaving aside the special case of algorithmic trading, market data are provided to users at different levels of quality of information (QoI) depending on their subscription; quality attributes in this context refer primarily to granularity, timeliness, and completeness, ranging from fine-granular tick data to end-of-day aggregations.

Traders, analysts, and other stakeholders utilize market data of their required quality in interactive decision support systems called market data terminals to identify investment opportunities for specific sets of symbols they are interested in. The Infront Professional Terminal (IPT) shown in Fig. 1 is an example of a terminal solution that fuses market data with metadata, news, interactive analytics, advanced visualizations, and direct trading functionality.

2 Data Set: One Week of Real Tick Data

The data used for this course project are based on a week's worth of real tick data captured by Infront Financial Technology GmbH in 2021. The full data set *Trading Data* used for the project is publicly available [1] licensed under an open license[1].

2.1 Full Data Set: *Trading ata*

The full data set contains 289 million events consisting of tick data and housekeeping events recorded from Monday, November 8th, to Sunday, November 14th, 2021. The data cover 5504 equities and indices that are traded on three European exchanges (exchange code in parentheses): Paris (FR), Amsterdam (NL), and Frankfurt/Xetra (ETR). All tick data events for security type *equities* (e.g., Royal Dutch Shell or Siemens Healthineers) and *indices* are contained in the set as they had been captured on these days by Infront's systems. Consequently, this data set reflects real update rates per second and can also be used for complex event processing research tasks that need to correlate events for indices with those for equities. In the case of exchange-specific and ambiguous identifiers for the same symbol, these have already been normalized by Infront, while global date and CEST timestamp information have been added together with various metadata for initial enrichment. Note that the total order of events per symbol across exchanges cannot be assumed.

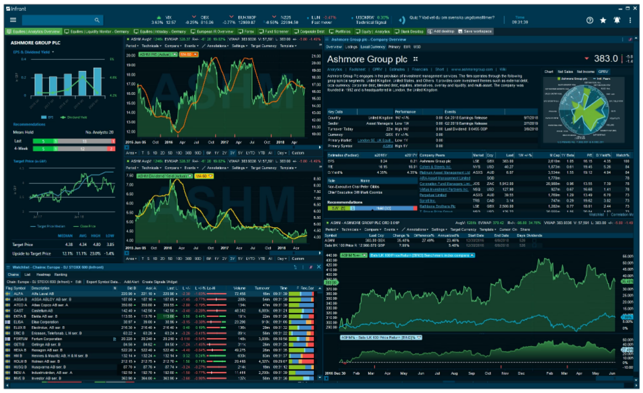

Figure 1: Example screen of a market data terminal product here: Infront Professional Terminal).

The distribution of events in the data set can be briefly summarized as follows: slicing by source, most traffic has been recorded from source Paris (55) followed by Amsterdam (28) and Frankfurt (17). Slicing by instrument type, we observe that more events are generated by activities regarding equities (239,5 million) than for indices (49,5 million). Actual trading activities (e.g., price events), however, are much higher for indices (49 million) than for equities (10 million). Furthermore, analysing the distribution of events across symbols reveals a long-tail distribution for general event types but also for trading activities regardless of exchange: most traffic is related to only a small number of symbols.

2.2 Attributes and Format

For convenience and portability, Trading Data is provided as a collection of flat comma-separated values (CSV) files (one file per day). Each line in a file represents a single event. Event types (e.g., *bid*, *ask*, *price*) are not explicitly marked but are identified by the respective non-NULL attributes (e.g., attribute *ask* for an *ask* event).

The attributes available in events of the data sets are shown in Table 1; The attributes directly relevant for this course project (CP) are marked in the third column with a star (★). Global CEST timestamps are in the format HH:MM:SS.ssss while dates are stored as DD-MM-YYYY.

Some events appear to come with no payload in the full data set. We had to balance the desire for a raw and unabridged data set

[1] http://creativecommons.org/licenses/by-nc-sa/4.0/

with the need to protect intellectual property and to keep the data set's size as small as possible. In particular, only a small subset of attributes needs to be evaluated in the course project. Hence, we eliminated certain attributes to preserve the number of events and their update patterns over time while reducing the overall size.

3 Project Requirements and Problem Definition

ID	Title	Description	CP
1	ID.[Exchange]	Unique identifier for this symbol with trading exchange: Paris (FR) / Amsterdam (NL) / Frankfurt (ETR)	
2	SecType	Security type: [E]quity or [I]ndex	
3	Date	System date last received update	
4	Time	System time last received update	
5	Ask	Price of best ask order	
6	Ask volume	Volume of best ask order	
7	Bid	Price of best bid order	
8	Bid volume	Volume of best bid order	
9	Ask time	Time of last ask	
10	Day's high ask	Day's high ask	
11	Close	Closing price (six digits)	
12	Currency	Currency (according to ISO 4217)	
13	Day's high ask time	Day's high ask time	
14	Day's high	Day's high (price)	
15	ISIN	ISIN (International Securities Identification Number)	
16	Auction price	Price at midday's auction	
17	Day's low ask	Lowest ask price of the current day	
18	Day's low	Lowest price of the current day	
19	Day's low ask time	Time of lowest ask price of the current day	
20	Open	First price of current trading day	
21	Nominal value	Nominal Value	
22	Last	Last trade price	
23	Last volume	Last trade volume	
24	Trading time	Time of last update (bid/ask/trade)	
25	Total volume	Cumulative volume for current trading day	
26	Mid price	Mid price (between bid and ask)	
27	Trading date	Date of last trade	
28	Profit	Profit	
29	Current price	Current price	
30	Related indices	Related indices	
31	Day high bid time	Time of day's highest bid	
32	Day low bid time	Time of day's lowest bid	
33	Open time	Time of open price	
34	Last price time	Time of last price	
35	Close time	Time of closing price	
36	Day high time	Time of day's high	
37	Day low time	Time of day's low	
38	Bid time	Time of last bid update	
39	Auction time	Time of last auction price	

Table 1: Attributes in *Trading ata*: syntax and semantics.

This course project requires students to implement a basic trading strategy commonly used by intraday traders in real life. The strategy involves implementing two specific queries and creating a visualization.

Query 1. Identifying trends in price movements for individual symbols using event aggregation over tumbling windows.
Query 2. Generating buy/sell advisories upon detecting specific patterns using complex event processing.
Smart Visualization. Students can also visualize the results of the queries in their projects.

The semantics of the above two queries and the corresponding visualizations of the results are provided below.

3.1 Definitions and Relaxations

We define the following terms and relaxations to allow students to focus on the actual engineering aspects of the course project:

(1) Each instrument instance is identified by a *symbol* $s \in S$ consisting of a unique string and the exchange code of the exchange the instrument instance is being traded on; e.g., the symbol *RDSA NL* denotes shares of *Royal Dutch Shell* traded on the Amsterdam exchange[2].
(2) All events have to be grouped by their symbol.
(3) All metrics for all $s \in S$ must constantly be calculated, but updates need to be provided only for a subset of symbols $s \in \overline{S} \quad S$ the platform subscribes to (simulating traders).
(4) Calculations are based on windows of 5 minutes' length.
(5) The first window w_0 starts at 0:00 (midnight) CEST.
(6) Windows do not overlap (tumbling windows).
(7) A window w is evaluated once the next window w_{+1} starts.

3.2 Quantitative Indicators Query 1)

The first query defines one of the most essential indicators for each symbol used in technical analysis to identify trends: the *exponential moving average* (EMA). Multiple price events observed within w minutes define a window of length w (e.g., $w \quad 5$ minutes). In our example a window of length w cannot be evaluated until the next window starts. The EMA of the current window is calculated by weighting the price last observed in the current window with the EMA of the previous window. Furthermore, we define $EM^{j}_{s,0} \quad 0$, i.e., for the first reading of a symbol s in the very first window, we assume the EMA of the previous window is zero.

$$EM^{j}_{s,w} \quad \Big[Close_{s,w} \cdot \quad \frac{2}{1+j}\Big)\Big] + \underbrace{EM^{j}_{s,w \quad 1}}_{\text{prev. window}} \cdot \Big[1 \quad \frac{2}{1+j}\Big)\Big]$$

with

$|w|$: window duration in minutes
j : smoothing factor for EMA with $j \in \{38, 100\}$
s : symbol $s \in S \quad \{s_1, \ldots, s_n\}$
$Close_{s,w}$: last price event for s observed in window w
EM^{j}_{s,w_0} 0

[2] https://www.xtb.com/int/trading-services/range-of-markets/shares-trading/rdsa-nl

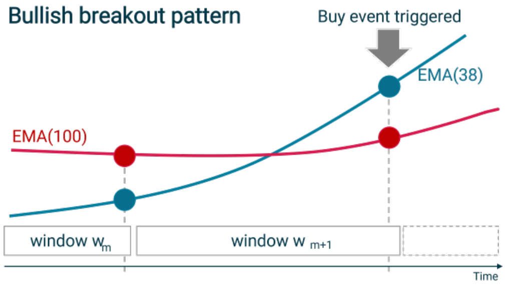

Figure 2: Crossover called a *bullish* breakout pattern.

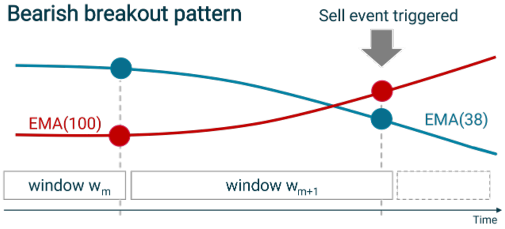

Figure 3: Crossover called a *bearish* breakout pattern

3.3 Breakout Patterns: Crossovers Query 2)

The quantitative indicators of Query 1 are used in Query 2: *breakout patterns* can be identified by tracking two EMAs per symbol that are computed over different intervals.

Generally, breakout patterns describe meaningful changes in the development of a price that indicate the start of a *trend* (even if only temporary). A change is called a *bullish breakout*, if the price is starting to rise steadily (crossover from below / breaking through the *support area*) and a *bearish breakout* if the price is going to lose steadily (crossover from above / breaking through the *resistance area*). Properly identifying such changes and their nature in a timely manner allows a trader to monetize this knowledge by immediately buying (in case of a bullish breakout) or selling (in case of a bearish breakout) to maximize revenue.

3 3 1 Bullish Breakout Pattern: Buy Advisory Generally, we detect a bullish breakout pattern for a symbol once the EMA with the shorter interval j_1 starts to overtake the EMA with the longer interval j_2. In this case, a *buy advise event* must be created immediately so that a trader can still benefit from a relatively low price.

For long intervals of j_1 50 days and j_2 100 days this crossover is specifically called a *golden cross* to indicate a golden opportunity for long-term investments.

For this project, we use a granularity of minutes by setting j_1 38 and j_2 100 and create a *buy advise event* upon detecting a crossover as illustrated in Fig. 2 and formalized in Equations 1 and 2. A bullish breakout pattern can be observed if and only if

$$EM^{38}_{s,w} > EM^{100}_{s,w} \text{ and} \tag{1}$$

$$EM^{38}_{s,w_{\;1}} \leq EM^{100}_{s,w_{\;1}} \tag{2}$$

Subsequently, a *buy advise* event must be generated.

3 3 2 Bearish Breakout Pattern: Sell Advisory Generally, we detect a bearish breakout pattern for a symbol once the EMA with the longer interval j_2 starts to overtake the EMA with the shorter interval j_1. In this case, a *sell advise event* must be created immediately so that a trader can still sell at a relatively high price.

Conversely to the golden cross described earlier, a bearish pattern for j_1=50 days and j_2=100 days is specifically called a *death cross*.

For the bullish pattern, we use a granularity of minutes by setting j_1 38 and j_2 100 and create a sell advise event as shown in Fig. 3 and formalised in Equations 3 and 4.

A bearish breakout pattern can be observed if and only if

$$EM^{38}_{s,w} \qquad EM^{100}_{s,w} \text{ and} \tag{3}$$

$$EM^{38}_{s,w_{\;1}} \geq EM^{100}_{s,w_{\;1}} \tag{4}$$

Subsequently, a *sell advice event* must be generated.

3.4 Smart Visualization

Whether users can fully exploit a decision support system for trading does not only depend on the correctness and performance of its implementation; being able to visually cut through the noise and visually emphasize the relevant data to the user is almost as important. Hence, students are encouraged to find a smart way to visualize the results of the queries for bonus points.

4 Submission Requirements

Students must implement a system to solve the problems (queries and visualization) described in Section 3. Each student is required to submit a link to the system's GitHub repository (ensure the repository is set to public visibility) along with a 4-5 page report detailing the system. For group projects, students must clearly specify the contributions of each member. The detailed requirements for the project report are as follows:

4.1 Project Report Requirements

Template. Use the ACM Proceeding Template: LaTeX[3] or Word [4].
Report Organization. The report should be 4-5 pages long and include the following sections. Students may use their own section titles, but the content should align with the following structure:

(1) **Abstract.**
(2) **Introduction.** Provide a brief background and overview of the project.
(3) **System Architecture.** Describe the design choices and motivations in detail.
(4) **Implementation.** Explain how the project was implemented, including the hardware/cloud services, programming tools, and GUI tools used.
(5) **Performance Evaluation.** Assess the system's performance, including its correctness (e.g., whether query results are accurate), scalability (how the system performs with varying resources, such as CPU, or workloads, such as events per second), and resource utilization (e.g., GPU and CPU usage).
(6) **Conclusion.**

[3] https://www.overleaf.com/latex/templates/acm-hypertext-conference-template/pchbkqfnmxgr
[4] https://www.acm.org/binaries/content/assets/publications/word_style/interim-template-style/interim-layout.docx

References

[1] S. Frischbier, J. Tahir, C. Doblander, A. Hormann, R. Mayer, and H.-A. Jacobsen. 2022. *DEBS 2022 Grand Challenge Data Set: Trading Data.* https://doi.org/10.5281/zenodo.6382482