

Analyzing Real Time Stock Market Data using Apache Flink

Amritpal Singh ^a, Aditya Khamparia ^b

^a *Lovely Professional University, Phagwara, Punjab, India*

^b *Babasaheb Bhimrao Ambedkar University, Uttar Pradesh, India*

Abstract

This research paper presents the study of Apache Flink which is unified stream processing and batch processing framework. Apache Flink is best suited for data analytics, event-driven and data pipeline applications. Flink has been shown to grow to hundreds of cores and gigabytes of application state, provide high throughput and low latency, and run some of the most demanding stream processing applications in the world. In this study, flink has been used to analyze real time stock market data and presented useful insights.

Keywords

Apache Flink, Batch Processing, Stream Processing, Dataset API, Datastream API

1. Introduction

With the rapid increase in data production and sophisticated data collection technologies, businesses face the task of making sense out of this mountain of raw data. Apache hadoop has evolved as the go-to platform for working with large data at the batch processing level. Until recently, as one search for architectures to create applications for stream processing in real-time, there has been a gap. These programs have been an important part of a great many businesses. Examples of that include tracking social networking to assess market reaction to every new product you include releasing, stock market analysis and forecasting the result of an election based on election-related posts sentiments. Because the stock market is one of the most important industries for investors, stock market price trend prediction is usually a popular issue for academics in both financial and technological fields. Apache Flink has developed as the market leaders 'framework of choice for creating these centralized, real-time, data processing systems. Apache Flink can be used for the following use cases:

- Event-driven Applications
- Data Analytics Applications
- Data Pipeline Applications

2. Survey of Related Works

Kim and Han used a combination of artificial neural networks (ANNs) and genetic algorithms (GAs) with feature discretization to construct a model for predicting stock price index. [1]. Qiu and Song have proposed a solution for determining the trend of the Japanese stock market based on an optimized artificial neural network model [2]. Hassan and Nath used the Hidden Markov Model (HMM) for stock market forecasting on four different airlines' stock values. They divide the model's states into four categories: open price, close price, maximum price, and lowest price. The strength of this paper is that it does not require expert knowledge to construct a prediction model [3]. Lee coupled the support vector machine (SVM) with a hybrid feature selection method to predict market trends.

WCES-2022: Workshop on Control and Embedded Systems, April 22 – 24, 2022, Chennai, India.

EMAIL: apsaggu@live.com (Amritpal Singh)

ORCID: 0000-0002-6783-0960 (Amritpal Singh)

© 2022 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

A subset of the NASDAQ Index from the Taiwan Economic Journal Database (TEJD) in 2008 was used in this study. [4]. Sirignano and Cont utilised a deep learning algorithm that was trained on a set of common financial market characteristics. All buy and sell records for all transactions, as well as cancellations of orders for around 1000 NASDAQ shares via the stock exchange's order book, were included in the dataset [5]. The goal of this study is to compare two of the most widely used and promising big data frameworks: Apache Flink and Apache Hive. Authors utilize the BigBench benchmark, which was created for Apache Hive, to compare these two frameworks [6]. In this paper, authors look into distance-based outliers in metric space, where an entity's status as an outlier is determined by the number of other entities in its vicinity. Authors made use of Apache Flink, which is considered as cutting-edge streaming analytics platform [7]. McNally et al. used RNN and LSTM to forecast the price of Bitcoin in a paper. The feature engineering aspect was improved using the Boruta algorithm [8]. Weng et al. used ensemble techniques to forecast short-term stock prices in their work. There are five sets of data in this study's dataset. These datasets were gathered using three open-source APIs and the TTR R package. The main contribution of this article is that it established an R-based platform for investors that does not require users to submit their own data but instead calls an API to retrieve data from an internet source [9]. In their paper, Kara et al. used ANN and SVM to forecast the movement of a stock price index. The data set they used spans the Istanbul Stock Exchange's history from January 2, 1997, to December 31, 2007. The complete record of parameter modification processes is the work's main strength [10]. Huang et al. used a fuzzy-GA model to perform the stock selection job in their work. As the investment universe, they used the key stocks of the 200 highest market capitalization listed on the Taiwan Stock Exchange. In addition, the annual accounting records data and stock returns for the years 1995 to 2009 were obtained from the Taiwan Economic Journal (TEJ) database at www.tej.com.tw/ [11]. Fischer and Krauss used long short-term memory (LSTM) to predict financial markets in their article. The dataset they used was Thomson Reuters' S&P 500 index constituents. From December 1989 through September 2015, they collected all month-end constituent listings for the S&P 500 [12]. Pimenta et al. used multi-objective genetic programming to create an automated investment strategy that they applied to the stock market in their article. [13]. Huang and Tsai employed a filter-based feature selection paired with a hybrid self-organizing feature map (SOFM) support vector regression (SVR) model to estimate Taiwan index futures (FITX) trend in their article. To boost the training efficiency, they separated the training samples into clusters. The authors proposed a complete model for stock market analysis that combined two unique machine learning techniques [14]. The authors developed a viable model for real-world investment operations that may create three fundamental signals for investors to consider. They also compared and contrasted a number of comparable methods. They did not, however, indicate the length of time or computing complexity of their works. Meanwhile, the absence of financial domain knowledge was an unavoidable difficulty in their work [15]. Hafezi et al. created a bat-neural network multi-agent system (BN-NMAS) to estimate stock price in their article. The data was provided by the Deutsche Bundesbank. They also used the Bat algorithm (BA) to optimise the weights of neural networks. The authors used flowcharts to show the overall design and logic of their system architecture [16]. Long et al. used a deep learning technique to forecast stock price movement in their research. The authors utilised a new model using a hybrid model produced by several types of neural networks, and this paper provides motivation for developing hybrid neural network structures [17]. In their study, Nekoeiqachkanloo et al. designed a methodology with two different stock investment methodologies. The advantages of their proposed solution are self-evident. To begin, it is a complete system that includes data pretreatment and two separate algorithms for recommending the finest investment segments. Second, the system includes a forecasting component that preserves the time series' characteristics [18].

Limited data-preprocessing techniques established and used is one of the key flaws observed in similar efforts. The majority of technical work is on developing prediction models. The bulk of the technical work is devoted to the creation of prediction models. They chose the features by making a list of all the topics that have been discussed in previous works, running them through the feature selection algorithm, and then selecting the features with the highest votes.

3. Stock Real-Time Data Processing

For a particular stock there are thousands of transactions going on every second. Some are selling that stock and some are buying that stock. These thousands buy-and-sell transactions make the price and volume of a stock fluctuate very quickly, tick-by-tick, even second-by-second. So capturing that live data which is changing in seconds is what is considered as real-time stock data. This data can include what is the price of a stock, volume of stock, et cetera parameters in every minute, or in every second, depending on the stock actually. Actively traded stocks can fluctuate dramatically in every second.

Generally, if one has to work with the live current stock data, then you have to get it from any third party provider. That third party will provide you its APIs, through which the current stock data at this second will be streamed to machine. Same like getting tweets from Twitter provided APIs, there are few websites which will provide their APIs, and can get live current stock data from them, by paying them some monthly or yearly subscription fees. But in this use case, I have simply collected the live stock data, stored it in a file, and then processed that file. There are 4 columns in it. First column is the date, second column is the timestamp, third column is the price of the stock at that timestamp, and fourth column is the volume of that stock at that timestamp. Table 1 shows the sample dataset used for data analysis.

Table 1: Sample Dataset

Date	Timestamp	Price	Volume
07/03/2022	08:00:00	106	348746
07/03/2022	08:00:00	105	331580
07/03/2022	08:00:00	105.5	352352
07/03/2022	08:00:00	106.5	347253
07/03/2022	08:00:00	105	330164
07/03/2022	08:00:01	106.5	332688
07/03/2022	08:00:01	107	343413
07/03/2022	08:00:02	107	337008
07/03/2022	08:00:02	107	346299
07/03/2022	08:00:02	105	342927
07/03/2022	08:00:05	107	354739
07/03/2022	08:00:07	105.5	335504
07/03/2022	08:00:08	104.5	340119
07/03/2022	08:00:08	104	345891

One can imagine the frequency of the fluctuations of price and volume of this stock from this data set only. See, in this timeframe of only 1 second, we got five rows. First, the price was 106 INR, and volume was 348,746 shares. Then within the same 1 second, price dropped by one rupee and it became to 105, and the volume to this number. Again, in the 1 second window only, the price dropped by 50 paise, it became to 105.5, and the volume increased by this number. Now imagine that if we have these many rows for 1 second only, then how much data would be there for whole day. Of course, it will be big.

4. Result and Discussion

In this research, Apache Flink has been used for analyzing and processing stock market data. The implementation has been done with 11th Generation Intel Core i5 and 16 GB RAM configuration.

Now comes up processing part. Each Flink program consists of same basic parts:

- Obtain an execution environment (Dataset/Datastream execution environment)
- Load/create the initial data (SocketTextStream, Kafka, Text File etc.)
- Specify transformation on this data (FlatMap, Map, Filter etc.)
- Specify where to put the results of computations (WriteAsCsv)
- Trigger the program execution (Execute)

From this data of only 4 columns, a lot of case studies can be executed. Stock exports can generate a lot of insights from this data; even can predict the fate of this stock in future. Out of many use case studies available, we are going to perform a case study which will basically generate an every minute report of these things. This is the requirement set; for every 1 minute, the report should show the max trade price, minimum trade price, maximum trade volume, minimum trade volume, and also the percent change in max trade price and max trade volume from the previous minute's price and volume, respectively. This is how the first report should look like.

Table 2: Generated Report

From Timestamp	End Timestamp	Current Window Max Price	Current Window Min Price	% change in Max Price	Current Window Max Vol.	Current Window Min Vol.	% change in Max Volume
07/03/2022:08:00:00	07/03/2022:08:00:58	107.0	101.5	0.00	354881	330164	0.00
07/03/2022:08:01:03	07/03/2022:08:01:59	103.0	101.0	-3.74	354948	330514	0.02
07/03/2022:08:02:00	07/03/2022:08:02:59	106.0	102.5	2.91	354834	332433	-0.03
07/03/2022:08:03:01	07/03/2022:08:03:56	107.5	105.5	1.42	354922	331050	0.02
07/03/2022:08:04:00	07/03/2022:08:04:59	107.0	104.5	-0.47	354851	330202	-0.02
07/03/2022:08:05:11	07/03/2022:08:05:51	106.0	104.5	-0.93	353795	227712	-0.30

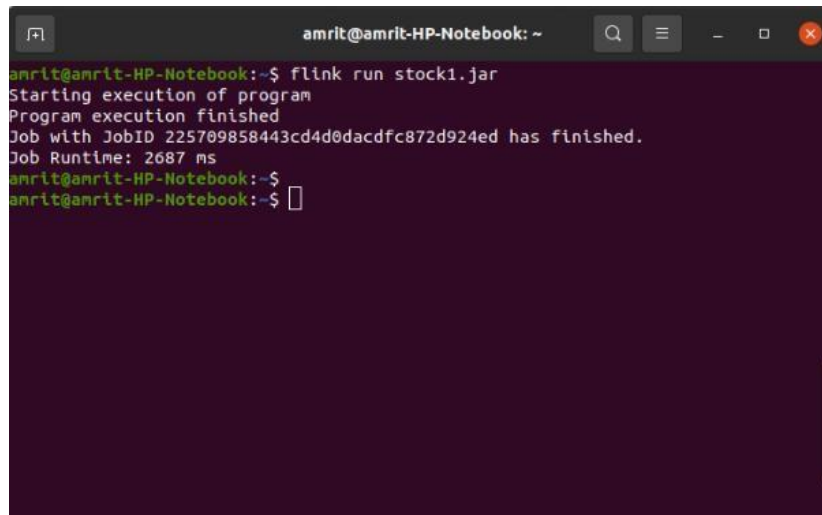
If the max trade price of this stock is changing by more than 5% in a 5-minutes window, then that event is to be recorded in that report. For example, let's suppose for a current window, from 0 to 5 minutes, the max trade price is 100 rupees, and in the second window, from 5.0 minutes to 10 minutes, the price is either going up by more than 5%, like it became to 106 or 107, or it decreased by 5%, like it becomes 93 or 92, then this event is to be recorded in a separate file with proper timestamp. Please note that this 5% change is to be calculated on max trade price, and not on the current price. So this is how the second report should look like.

Table 2: Alert Report

% Change	Previous Window Max Price	Current Window Max Price	Event Timestamp
-7.83%	115.0	106.0	07/03/2022:09:20:00
10.28%	107.0	118.0	07/03/2022:09:34:48
7.00%	121.5	130.0	07/03/2022:09:48:02

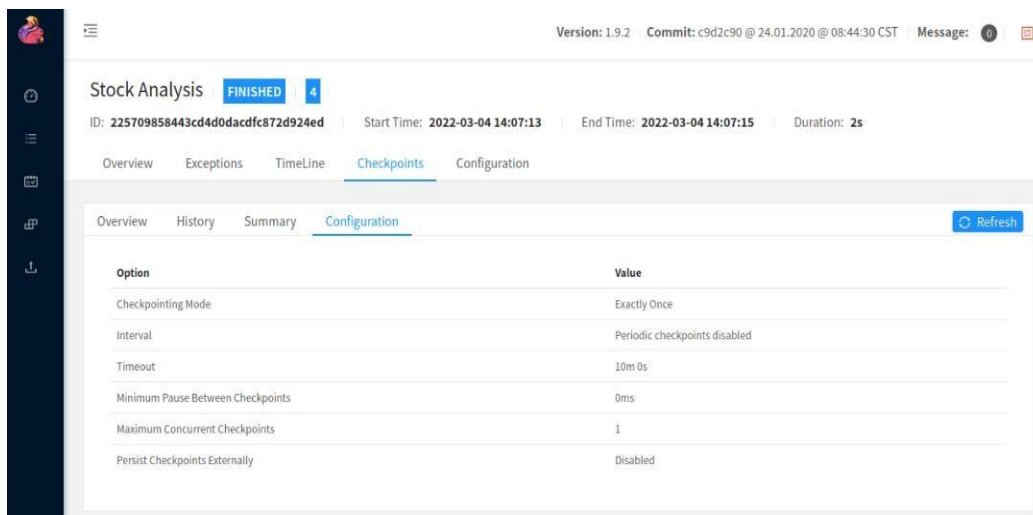
Large change detector of how much percent, what was the previous window max price, and what is the current window max price, and this even got captured at what timestamp. For complete data, we got these many alerts. First alert was at 9:20 when the max price dropped by 7.83%, then at 9:34 we got second alert, but this time the max price increased by 10.28%. So basically there are two reports to be generated for this case study. From this report, data science expert can find out patterns of the

changes, which will help to generate insights for this stock. These insights can turn out to be a huge profit gain for the stock. Figure 1 shows the flink job in action. Figure 2 shows the flink configuration parameters which include checkpointing mode, interval, timeout, minimum pause between checkpoints, maximum concurrent checkpoints and persist checkpoints. Figure 3 show the directed acyclic graph generated as part of flink processing.



```
amrit@amrit-HP-Notebook: ~  
amrit@amrit-HP-Notebook:~$ flink run stock1.jar  
Starting execution of program  
Program execution finished  
Job with JobID 225709858443cd4d0dacdfc872d924ed has finished.  
Job Runtime: 2687 ms  
amrit@amrit-HP-Notebook:~$  
amrit@amrit-HP-Notebook:~$
```

Figure 1: Execution of Flink Job



Version: 1.9.2 Commit: c9d2c90 @ 24.01.2020 @ 08:44:30 CST Message: 0

Stock Analysis **FINISHED** 4

ID: 225709858443cd4d0dacdfc872d924ed | Start Time: 2022-03-04 14:07:13 | End Time: 2022-03-04 14:07:15 | Duration: 2s

Overview Exceptions TimeLine **Checkpoints** Configuration

Overview History Summary **Configuration** Refresh

Option	Value
Checkpointing Mode	Exactly Once
Interval	Periodic checkpoints disabled
Timeout	10m 0s
Minimum Pause Between Checkpoints	0ms
Maximum Concurrent Checkpoints	1
Persist Checkpoints Externally	Disabled

Figure 2: Stock Analysis

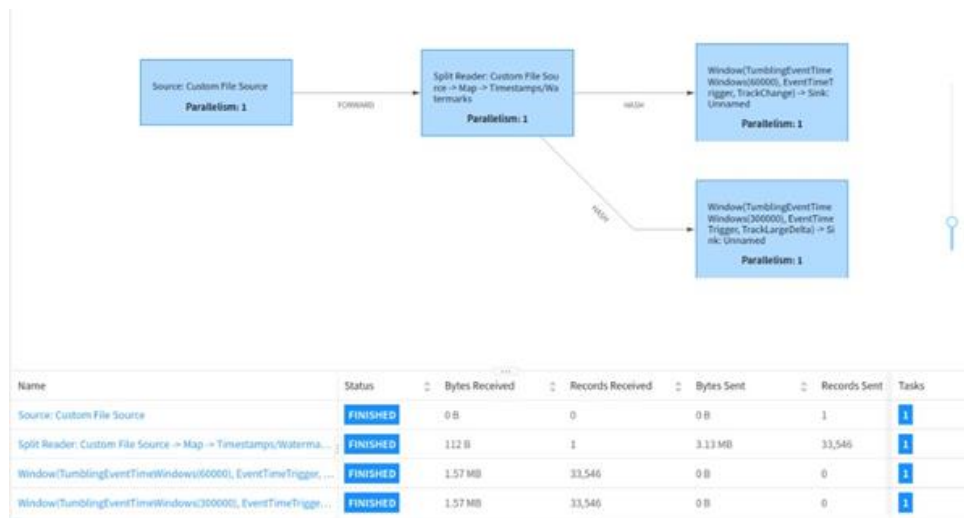


Figure 3: Directed Acyclic Graph

The study presented above can be utilized to comprehend a stock's short- and long-term behaviour. Depending on the risk appetite of the investor, a decision support system can be constructed to choose which stock to pick from the industry for low-risk low gain or high-risk big gain.

5. References

- [1] Kim K, Han I. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Syst Appl.* 2000;19:125–32. [https://doi.org/10.1016/S0957-4174\(00\)00027-0](https://doi.org/10.1016/S0957-4174(00)00027-0).
- [2] Qiu M, Song Y. Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLoS ONE.* 2016;11(5):e0155133.
- [3] Hassan MR, Nath B. Stock market forecasting using Hidden Markov Model: a new approach. In: *Proceedings—5th international conference on intelligent systems design and applications 2005, ISDA'05.* 2005. pp. 192–6. <https://doi.org/10.1109/ISDA.2005.85>.
- [4] Lei L. Wavelet neural network prediction method of stock price trend based on rough set attribute reduction. *Appl Soft Comput J.* 2018;62:923–32. <https://doi.org/10.1016/j.asoc.2017.09.029>.
- [5] Sirignano J, Cont R. Universal features of price formation in financial markets: perspectives from deep learning. *Ssrn.* 2018. <https://doi.org/10.2139/ssrn.3141294>.
- [6] Sonia Bergamaschi, Luca Gagliardelli, Giovanni Simonini, Song Zhu, BigBench Workload Executed by using Apache Flink, *Procedia Manufacturing*, Volume 11, 2017, Pages 695-702, ISSN 2351-9789, <https://doi.org/10.1016/j.promfg.2017.07.169>.
- [7] Theodoros Toliopoulos, Anastasios Gounaris, Kostas Tsihlias, Apostolos Papadopoulos, Sandra Sampaio, Continuous outlier mining of streaming data in flink, *Information Systems*, Volume 93, 2020, 101569, ISSN 0306-4379, <https://doi.org/10.1016/j.is.2020.101569>.
- [8] McNally S, Roche J, Caton S. Predicting the price of bitcoin using machine learning. In: *Proceedings—26th Euromicro international conference on parallel, distributed, and network-based processing, PDP 2018.* pp. 339–43. <https://doi.org/10.1109/PDP2018.2018.00060>
- [9] Weng B, Lu L, Wang X, Megahed FM, Martinez W. Predicting short-term stock prices using ensemble methods and online data sources. *Expert Syst Appl.* 2018;112:258–73. <https://doi.org/10.1016/j.eswa.2018.06.016>.
- [10] Kara Y, Acar Boyacioglu M, Baykan ÖK. Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul Stock Exchange. *Expert Syst Appl.* 2011;38(5):5311–9. <https://doi.org/10.1016/j.eswa.2010.10.027>.
- [11] Huang CF, Chang BR, Cheng DW, Chang CH. Feature selection and parameter optimization of a fuzzy-based stock selection model using genetic algorithms. *Int J Fuzzy Syst.* 2012;14(1):65–75. <https://doi.org/10.1016/J.POLYMER.2016.08.021>.

- [12] Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions. *Eur J Oper Res.* 2018;270(2):654–69. <https://doi.org/10.1016/j.ejor.2017.11.054>.
- [13] Pimenta A, Nametala CAL, Guimarães FG, Carrano EG. An automated investing method for stock market based on multiobjective genetic programming. *Comput Econ.* 2018;52(1):125–44. <https://doi.org/10.1007/s10614-017-9665-9>.
- [14] Huang CL, Tsai CY. A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. *Expert Syst Appl.* 2009;36(2 PART 1):1529–39. <https://doi.org/10.1016/j.eswa.2007.11.062>.
- [15] Thakur M, Kumar D. A hybrid financial trading support system using multi-category classifiers and random forest. *Appl Soft Comput J.* 2018;67:337–49. <https://doi.org/10.1016/j.asoc.2018.03.006>.
- [16] Hafezi R, Shahrabi J, Hadavandi E. A bat-neural network multi-agent system (BNNMAS) for stock price prediction: case study of DAX stock price. *Appl Soft Comput J.* 2015;29:196–210. <https://doi.org/10.1016/j.asoc.2014.12.028>.
- [17] Long W, Lu Z, Cui L. Deep learning-based feature engineering for stock price movement prediction. *Knowl Based Syst.* 2018;164:163–73. <https://doi.org/10.1016/j.knosys.2018.10.034>.
- [18] Nekoeiqachkanloo H, Ghogh B, Pasand AS, Crowley M. Artificial counselor system for stock investment. 2019. ArXiv Preprint arXiv:1903.00955.