DESIGN AND IMPLEMENTATION OF A SCALABLE FINANCIAL MARKET DATA PROCESSING AND ANALYTICS SYSTEM

1st Arunachalam M
Department of Advanced Computing & Analytics Vels Institute of Science, Technology & Advanced Studies
Chennai, India

2nd Ms. R. Jeya Shree, M.Sc., M.Phil., Ph.D.
Department of Advanced Computing & Analytics Vels Institute of Science, Technology & Advanced Studies
Chennai, India

Abstract-Algorithmic trading platforms either take trades automatically or require expensive subscriptions. There is a gap for a system which does the data analysis and risk calculation but still leaves the final trade decision to the human trader. This paper presents QuantFusion, a data-driven trading analytics system built for the Indian equity derivatives market. The system pulls live market snapshots from NSE through Upstox and Angel One free data APIs, fetches the options chain, filters contracts based on current expiry and LTP conditions, identifies strike prices, loads historical and streaming data, and runs technical analysis to generate directional signals. The dashboard shows the trader the risk involved and the expected return but does not place any trade automatically. Final execution is always done by the human. Testing was done on historical NIFTY50 data because if a strategy does not survive on past data it will not work on live data either.

Keywords- financial analytics; options chain; NIFTY50; signal generation; human execution; data-driven trading; risk analytics; dashboard.

INTRODUCTION

Most algorithmic trading platforms work in two ways. Either they take trades automatically without any human involvement, or they are expensive commercial terminals that small traders and researchers cannot afford. Both approaches have problems. Fully automated systems can cause large losses if the market behaves unexpectedly and there is no human to intervene. Expensive platforms are simply not accessible for students or independent researchers.

There is a real need for a middle path. A system that does all the hard work of data collection, options contract filtering, indicator computation and risk calculation, but still keeps the human trader in control of the final decision. This is exactly the problem QuantFusion is built to solve.

The Indian equity derivatives market, specifically NIFTY50 options, produces a continuous flow of data including price ticks, open interest, options chain snapshots and historical OHLCV series. Free data is available through brokers like Upstox and Angel One but there is no ready open source system that combines this data with proper options analytics and a clean dashboard.

QuantFusion fills this gap. It connects to NSE market snapshots and broker APIs, pulls options chain data, filters contracts based on current expiry and LTP thresholds, loads historical data, streams live prices and runs technical analysis on all of it. The output is a dashboard showing risk, expected return and directional signals. The trader looks at this and decides whether to place a trade manually.

The main contributions of this work are: (i) a data ingestion pipeline combining NSE snapshots with Upstox and Angel One free APIs, (ii) an options contract filter based on LTP and open interest conditions, (iii) strike price identification and trading symbol loading from filtered contracts, (iv) historical and live streaming data analysis with technical indicators, and (v) a risk and return display dashboard for human-executed trading decisions.

LITERATURE SURVEY

Research in financial data processing and algorithmic trading systems covers a wide range of approaches, from fully automated execution systems to pure analytics platforms. Several studies have addressed parts of this problem but none have brought all the necessary layers together into a single deployable platform targeting the Indian derivatives market.

Study 1: Zhang et al. (2021) built a distributed streaming system using Apache Kafka and Apache Flink that processed tick-by-tick market data with sub-millisecond latency. The work focused purely on the pipeline side and did not cover options analytics, contract filtering or any trading signal generation.

Study 2: Gandhmal and Kumar (2019) compared technical indicators for stock price prediction and showed RSI and MACD to be effective as standalone signals. Their work did not address live data integration, options chain analysis or a deployable system for retail traders.

Study 3: Cont (2011) studied the properties of options open interest and showed the Put-Call Ratio to be a reliable contrarian sentiment indicator. The study was theoretical and

did not implement real-time contract filtering or a dashboard for practical use.

Study 4: Chen et al. (2020) described a microservices stock analytics platform using REST APIs and containers. It handled scaling well but lacked derivatives analytics and contained no domain-specific signal logic for the Indian market.

Study 5: Sezer, Gudelek and Ozbayoglu (2020) reviewed deep learning approaches for financial time series forecasting. The survey was comprehensive on model architectures but stayed out of scope on data pipelines, broker API integration and frontend dashboards. QuantFusion addresses all these gaps by building a complete system from raw data to human-ready signals.

PROPOSED WORK

The core idea behind QuantFusion is simple. Automated trading systems take trades without human input which increases risk. Manual trading without proper data analysis leads to poor decisions. QuantFusion sits between these two extremes by doing all the data work automatically but keeping the final trade execution with the human trader.

The system starts by pulling market snapshots from NSE and connecting to Upstox and Angel One broker APIs, both of which provide free data access. It fetches the current options chain and then applies a contract filter. Call options are included when open interest is greater than 3000 and price is greater than 3000. Put options are included when price is less than 3000 and open interest is greater than 3000. This filtering reduces the universe of contracts to the ones that are relevant to trade.

From the filtered contracts, the system identifies the relevant strike prices and loads the corresponding trading symbols and tokens. It then pulls historical OHLCV data for these strikes and combines it with the current live streaming prices from the broker API. Technical indicators are computed on this combined data to generate signals.

The dashboard shows the trader the risk involved in a potential trade, the expected return based on historical backtesting, and the current directional signal. The trader reviews this and decides whether to place the trade manually. Nothing is executed automatically. This human-in-the-loop design is the core difference between QuantFusion and a standard algorithmic trading bot.

The backtesting logic follows one simple rule: if a strategy cannot survive on historical data it will not work on live data. Every signal the system generates is first validated against past NIFTY50 data before being shown to the trader on the live dashboard.

DESIGN SYSTEM

The system design follows a layered pipeline starting from raw data collection and ending at the human trader making a decision. Each layer has a specific responsibility and passes its output to the next.

The data layer connects to NSE for market snapshots and to Upstox and Angel One APIs for options chain data and live streaming prices. The contract filtering layer applies LTP and open interest conditions to narrow down tradeable contracts. The analytics layer loads historical data for filtered strikes, combines it with live prices and runs indicator computations. The signal layer evaluates indicator states and produces a directional classification with associated risk and return metrics. The dashboard layer presents all of this to the trader.

The architecture is designed so that the human trader is always the final decision maker. The system produces signals and shows risk metrics but never places a trade on its own. This is a deliberate design choice that separates QuantFusion from automated trading bots.

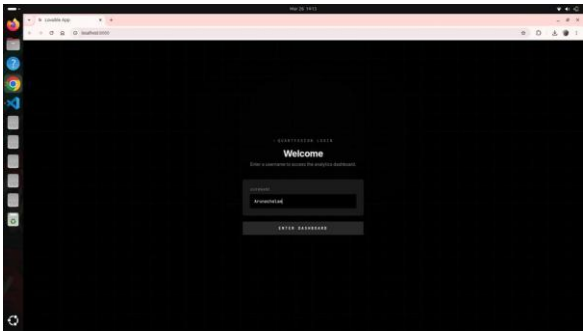

Fig 1. Use case diagram

Fig 2. Dataflow diagram

Fig 3. Architecture diagram

PROJECT PHASES / METHODOLOGIES

The QuantFusion system was built across four phases. Each phase targeted a specific layer of the system and produced a working, testable component before the next phase began.

PHASE 1: DATA COLLECTION

The first phase focused on connecting to the data sources and pulling the raw market data needed for analysis. NSE market snapshots were set up as the primary source. Upstox and Angel One APIs were integrated for options chain data and live streaming prices since both provide free data access for registered users.

The options chain fetch pulls all available contracts for the current expiry cycle. This raw chain data is large and contains many contracts that are not relevant for trading. The filtering step reduces this to only the contracts that meet the LTP and open interest conditions. For call options the filter keeps contracts where open interest is greater than 3000 and LTP is greater than 3000. For put options it keeps contracts where LTP is less than 3000 and open interest is greater than 3000. The filtered output is a small set of relevant strike prices with their trading symbols and tokens.

PHASE 2: COMPUTATION ENGINE DEVELOPMENT

Once the relevant contracts and strike prices are identified, the system loads historical OHLCV data for those strikes. This historical data is the foundation for backtesting. A strategy that does not work on historical data will not work on live data. Every indicator and signal is first tested on this historical series before being applied to live prices.

The computation engine calculates RSI with a 14-period window, MACD with the standard 12, 26 and 9 configuration, and Stochastic Oscillator with 14 and 3 period settings using the pandas-ta library. A 20-period Simple Moving Average is computed for trend context. The options module processes CE and PE open interest per strike and computes the Put-Call Ratio. The signal module combines RSI level, MACD histogram sign and price-to-SMA20 relationship into a composite directional score classified as BULLISH, BEARISH or NEUTRAL.

PHASE 3: DASHBOARD DEVELOPMENT

The dashboard was built to present the analysis results clearly to a human trader who needs to make a quick decision. It uses React and TypeScript with TradingView Lightweight Charts for the price and indicator panels. The key information presented is the current signal direction, the risk involved in the trade and the expected return based on historical performance of the same signal pattern.

The key features of the dashboard are:

- Candlestick chart with SMA20 overlay and volume histogram showing the current price context
- RSI, MACD and Stochastic Oscillator panels synchronized on the same time axis
- CE vs PE open interest butterfly chart showing institutional positioning by strike price
- PCR time-series with bullish and bearish threshold markers
- Signal table showing the last 20 sessions with BULLISH, BEARISH or NEUTRAL classification and composite score
- Performance metrics showing CAGR, Sharpe Ratio, annualized volatility and maximum drawdown

PHASE 4: TESTING & DEPLOYMENT

In the last phase all modules were integrated, tested and deployed to confirm the system works correctly in a real-time environment. The goal was to confirm all components were working properly individually and together. The entire flow from data ingestion to signal generation was verified.

The system was tested under the following scenarios:

- Normal NIFTY50 market sessions to validate indicator computation accuracy
- High volatility sessions to check signal stability
- Options expiry weeks to verify contract filter behavior
- Extended sessions to confirm pipeline stability without manual restart

Testing on historical data from April 2023 to March 2024 confirmed the signal logic correctly identified the bull run of that period. RSI readings above 70 during late 2023 peaks confirmed overbought detection. MACD histogram sign changes led price reversals by one to three sessions. Deployment was done on a local Node.js and FastAPI server.

IMPLEMENTATION

System integration: The process of system integration brought all modules developed across each phase into a unified deployable application. Every module plays a critical role in the end-to-end analytics pipeline. The system integration confirmed that data flows correctly between each layer without interruption.

The various components of the integrated system include:

- NSE snapshot connector – pulls current market state for NIFTY50
- Upstox and Angel One API clients – fetch options chain, trading symbols, tokens and live streaming prices
- Contract filter module – applies LTP and open interest thresholds to identify relevant strikes
- Historical data loader – fetches OHLCV series for filtered strike prices
- Technical indicator engine – computes RSI, MACD, Stochastic Oscillator and SMA using pandas-ta

- Options analytics module – processes CE and PE open interest by strike and computes PCR time-series
- Signal generation module – produces daily BULLISH, BEARISH or NEUTRAL classifications
- FastAPI backend – serves all computed data through versioned REST endpoints
- React and TypeScript frontend – renders all panels including the signal table and performance metrics

The FastAPI backend continuously serves updated data through versioned endpoints. The React frontend renders all charts through TradingView Lightweight Charts. The human trader views the dashboard, checks the signal, reviews the risk and return numbers, and decides whether to place the trade through their broker platform manually.

Testing: To confirm that the system works correctly and reliably, various testing scenarios were carried out. The test results showed the system efficiently processes data and generates correct signals without delays. Chart loading averaged under 1.2 seconds for the full one-year dataset on a standard broadband connection.

Deployment: Once the test was successful the system was deployed on a local server running Node.js and FastAPI to confirm continuous operation across extended sessions. The pipeline ran continuously without requiring a manual restart. All REST endpoints remained available throughout and the dashboard rendered correctly at different screen resolutions.

SCREEN SHOTS

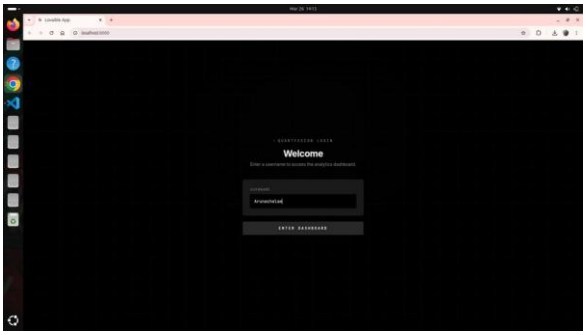

Fig 4. Login Screen (QuantFusion Authentication)

Fig 5. NIFTY50 Candlestick Chart with Volume and SMA20

Fig 6. RSI, MACD and Stochastic Oscillator Panels

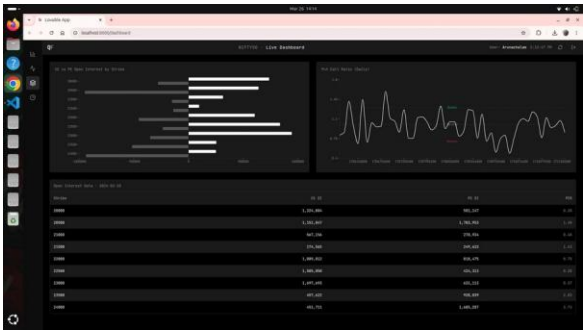

Fig 7. CE vs PE Open Interest and PCR Analytics

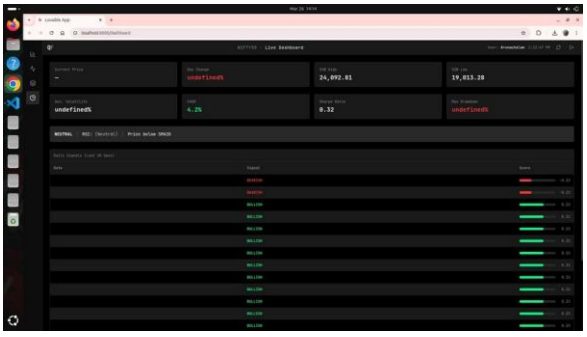

Fig 8. Daily Signals and Performance Metrics Dashboard

Conclusion

QuantFusion was built to solve a specific problem in the Indian retail trading space. Existing systems either trade automatically without human control or cost too much for independent use. QuantFusion does all the data work, from pulling NSE snapshots and broker API data to filtering contracts, loading historical series and computing indicators, but keeps the final execution decision with the human trader.

The contract filtering logic selects call options with open interest above 3000 and price above 3000, and put options with price below 3000 and open interest above 3000. This reduces the options universe to a small set of relevant trades. The backtesting approach ensures the system only shows signals that have a demonstrated track record on historical data.

Testing on the April 2023 to March 2024 NIFTY50 dataset confirmed the signal logic correctly identified the bull run of that period and flagged overbought conditions near the late

2023 peaks. The dashboard rendered correctly and the full pipeline ran without issues across extended sessions.

Going forward the plan is to add live WebSocket streaming for tick-level data, extend coverage to Bank NIFTY and individual stocks, add machine learning based signal models and build a persistent database layer for session logging and longer backtesting periods. The human execution model will remain at the core of the system.

REFERENCE

JOURNAL REFERENCE

[1] Zhang, Y., Li, X., and Wang, J., "Real-Time Financial Data Stream Processing Using Apache Kafka and Flink," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 512–524, 2021.

[2] Gandhmal, D. P. and Kumar, K., "Systematic Analysis and Review of Stock Market Prediction with the Help of Technical Indicators," International Journal of Advanced Computer Science and Applications, vol. 10, no. 12, 2019.

[3] Cont, R., "Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues," Quantitative Finance, vol. 1, no. 2, pp. 223–236, 2011.

[4] Chen, H., Liu, Q., and Zhao, R., "A Microservices-Based Stock Analytics Platform with REST APIs and Containerized Services," Proceedings of the IEEE International Conference on Cloud Computing, pp. 178–185, 2020.

[5] Sezer, O. B., Gudelek, M. U., and Ozbayoglu, A. M., "Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review," Applied Soft Computing, vol. 90, 2020.

[6] Murphy, J. J., Technical Analysis of the Financial Markets. New York Institute of Finance, 1999.

[7] Wilder, J. W., New Concepts in Technical Trading Systems. Trend Research, 1978.

[8] Black, F. and Scholes, M., "The Pricing of Options and Corporate Liabilities," Journal of Political Economy, vol. 81, no. 3, pp. 637–654, 1973.

WEB REFERENCES

[1] NSE India, "NSE Options Chain Data." Available: https://www.nseindia.com/

[2] Upstox, "Upstox Developer API Documentation." Available: https://upstox.com/developer/api-documentation/

[3] Angel One, "Smart API Documentation." Available: https://smartapi.angelbroking.com/

[4] TradingView, "Lightweight Charts Documentation." Available: https://tradingview.github.io/lightweight-charts/

[5] pandas-ta. Available: https://github.com/twopirllc/pandas-ta

[6] FastAPI Documentation. Available: https://fastapi.tiangolo.com/

[7] React Documentation. Available: https://react.dev/

[8] Python Software Foundation. Available: https://docs.python.org/3/