

LARGE LANGUAGE MODEL AGENTS AS INVESTMENT ADVISORS: A BACKTESTED COMPARISON OF GPT-3.5, GPT-4 AND GPT-4-TURBO OVER INDIAN STOCK DATA

Ankit Tripathi¹, Michaela Veselá¹, Oldrich Trenz¹

¹Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic

DOI: https://doi.org/10.11118/ecos2026_039

ABSTRACT

This article asks whether an OpenAI GPT model can act as a conversational investment advisor when it reads structured stock data instead of free text. A pandas DataFrame of daily prices for Reliance Industries (RELIANCE.NS), spanning 1 April 2019 to 28 March 2024, was enriched with the signals of four hard-coded technical strategies: gap trading, trend following, Fibonacci retracement and mean reversion. Each strategy was backtested with Pyfolio, and three LangChain DataFrame agents wrapping GPT-3.5-Turbo, GPT-4 and GPT-4-Turbo answered the same seven plain-English questions. Trend following gave the highest backtest return (annual 13.60%, cumulative 86.60%), while Fibonacci retracement was the least risky (max drawdown only -1.10%, Omega ratio 6.33). The GPT-4-Turbo agent gave the most context-aware answers; GPT-3.5-Turbo was inconsistent and once misread a signal. The results hold for a single asset and a no-cost in-sample backtest, so they describe a feasibility study rather than live trading performance.

Keywords: large language models, GPT agents, algorithmic trading, technical analysis, backtesting, Indian stock market

JEL Code: C58

1 INTRODUCTION

A beginner who opens a brokerage account faces two questions at once: which trading rule to follow and which tool to trust for advice. Generative artificial intelligence has reached finance quickly. JPMorgan filed for a service named IndexGPT, and surveys now track a whole family of finance-specific large language models (Lee *et al.*, 2024; Li *et al.*, 2023). What stays unclear is



This article is licensed under CC BY-SA 4.0 Attribution-ShareAlike 4.0 International license.
<https://creativecommons.org/licenses/by-sa/4.0/deed.en>

whether a general-purpose model, handed nothing but one company's price history, can turn that data into advice a non-expert can act on.

This work builds such a tool and tests it. Three GPT agents read a single stock's daily prices, already marked up with the entry and exit signals of four technical strategies, and answer ordinary investment questions. The setup is deliberately narrow. It is an offline backtest on one asset paired with a qualitative comparison of three models, not a live trading record and not a benchmark of model accuracy in the abstract.

A second motive is geographic. Most published work on algorithmic and LLM-based trading focuses on larger and more frequently studied markets, while the Indian market remains less represented in this specific research stream. A Scopus search reported in the source thesis returned 445 documents for "Trading in Indian stock market", 15 documents for "Algorithmic trading in Indian stock market" and no records for the specific queries "FinLLMs in Indian stock market" and "FinPLMs in Indian stock market".

1.1 Large language models in finance

The transformer architecture (Vaswani *et al.*, 2023) and the pre-trained encoders that followed it (Devlin *et al.*, 2019) made it practical to apply general language models to specialised domains. Recent surveys map this move into finance and list both the opportunities and the failure modes, hallucination chief among them (Lee *et al.*, 2024; Li *et al.*, 2023; Kalyan, 2024). Most of these systems read unstructured text: news, filings, social posts. Whether a model can instead read clean tabular data and reason over it is the question taken up here, in line with the agent-based view of Wang *et al.* (2024) and the deep-learning trading models of Pan *et al.* (2024).

1.2 Technical trading strategies

The four strategies were chosen because each rests on published evidence and each reacts to a different market regime. Overnight gap behaviour is documented by Akbas *et al.* (2022) and Ho *et al.* (2023). Trend following through moving averages has a long empirical record (Eggebrecht and Lütkebohmert, 2023; Zakamulin and Giner, 2020). Mean reversion and the combination of technical rules with portfolio construction round out the set (Macedo *et al.*, 2017; Banik *et al.*, 2022). The point is not to find one winning rule but to see how the agent's reason across several, so that the advice can weigh return against risk rather than chase a single number.

2 METHODOLOGY

Daily OHLCV data for Reliance Industries Limited (ticker RELIANCE.NS on the National Stock Exchange of India) were pulled from Yahoo Finance for 1 April 2019 to 28 March 2024 and exported to CSV. The five-year window holds 1235 trading days. After the strategy signals and technical indicators were added, the agents operated on a DataFrame of 1235 rows and 27 columns.

The work was done in Google Colaboratory using Python with pandas (pinned to 1.5.3 for Pyfolio compatibility), NumPy, Matplotlib, the ta technical-analysis library and Pyfolio for the risk and return tear sheets. Four strategies were coded as columns of buy and sell signals and then each was backtested separately:

Gap trading acts on overnight returns: it buys when the opening gaps down between -1% and -2% and sells one day later when the signal clears. Trend following uses two simple moving averages, a 10-day fast line and a 50-day slow line in the final implementation, and trades on their crossover. Fibonacci retracement levels were calculated from the maximum and minimum closing prices in the sample and combined with MACD based on 10-day and 30-day EMAs and a 12-day signal line. **Mean reversion** pairs Bollinger Bands (20-day mean, ± 2 standard deviations) with a 6-day RSI, buying when RSI falls below 30 below the lower band and selling when RSI passes 70 above the upper band.

Three agents were built with LangChain's experimental pandas DataFrame agent, each wrapping a different OpenAI model: Agent1 = GPT-3.5-Turbo, Agent2 = GPT-4, Agent3 = GPT-4-Turbo. Temperature was set to 0 to suppress randomness and reduce the chance of fabricated output. For a user query, the agent assembles the relevant indicators, recent price moves and strategy signals into a prompt and the model returns a plain-language answer (Figure 1).

Evaluation used seven predefined natural-language queries: dated buy or sell recommendations, an explanation aimed at a five-year-old in exactly ten sentences, a Fibonacci explanation for an experienced investor, and single-strategy versus all-strategy advice. The three agents' answers were compared query by query for accuracy, use of context, personalisation and any sign of hallucination. The choice of models followed third-party benchmarks current at the time (a MindsDB comparison from March 2024 and the LMSys Chatbot Arena leaderboard from April 2024), with GPT-3.5-Turbo kept as the cheaper baseline.

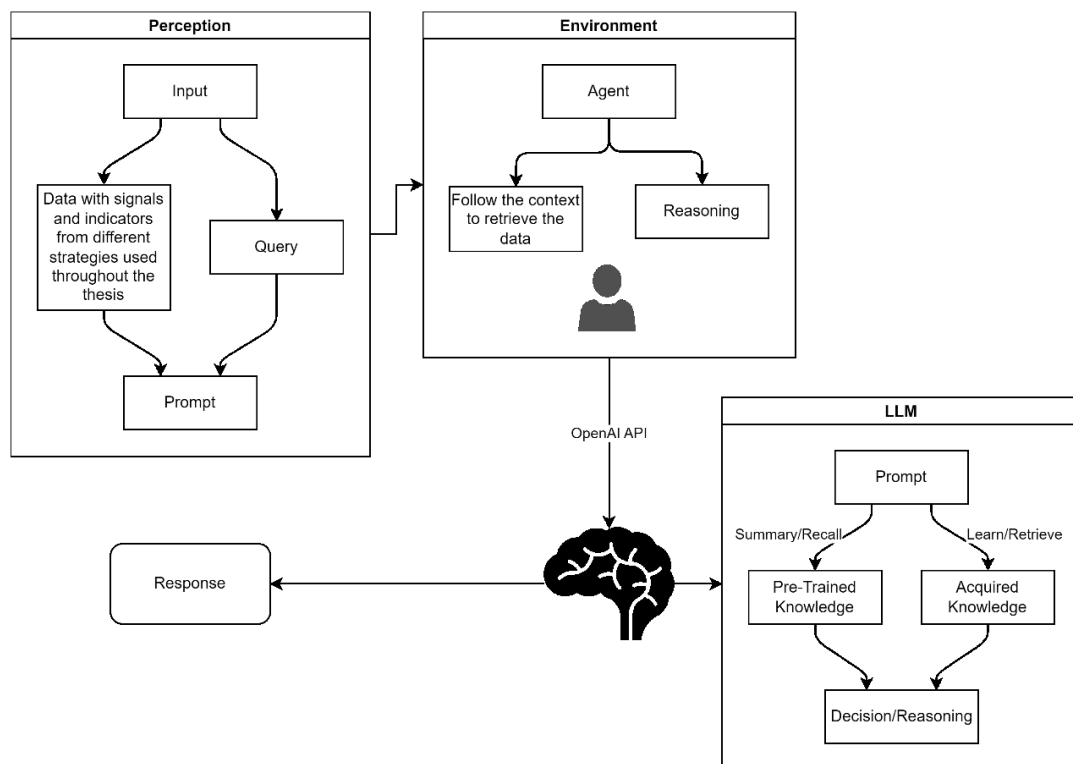


Fig. 1: Architecture of the LLM-based trading agent, showing the perception, environment and large language model layers communicating through the Open AI API (adapted from Tripathi, 2024)

3 RESULTS

Each strategy produced a different number of trades over the window: gap trading generated 48 buy and 48 sell signals, trend following generated 15 buy and 14 sell signals, Fibonacci retracement generated 7 buy and 6 sell signals, and mean reversion produced the largest number of short-term signals; the detailed entry-exit table in the source thesis reports 34 complete buy-sell pairs. This value should be checked against the original notebook before final submission. Table 1 reports the consolidated Pyfolio metrics; trend following is shown in bold as the highest-return variant (Tripathi 2024).

The strategies split cleanly by what they optimise. Trend following earned the most (annual 13.60%, cumulative 86.60%, Sharpe 0.79) but carried the highest volatility (18.10%) and the deepest drawdown (-26.80%, reached in early 2023). Fibonacci retracement earned little (2.50% annual) yet was the steadiest and safest: drawdown stayed within -1.10%, with the best Omega (6.33), Calmar (2.24) and Sortino (3.81) of the four. Mean reversion sat in the middle, with a moderate 3.50% annual return and a contained -9.50% drawdown. Gap trading was the only loser, at -1.60% annual and -7.80% cumulative, with negative risk-adjusted ratios across the board.

The agent comparison was qualitative, over the seven queries. Agent3 (GPT-4-Turbo) gave the strongest answers: it used the available indicators more consistently, addressed the full intent of the questions and is therefore the preferred model for the proposed proof-of-concept tool. Agent2 (GPT-4) was accurate to the strategy signals but less aware of the surrounding indicators, satisfactory rather than strong. Agent1 (GPT-3.5-Turbo) was the weakest: on one query it read a trend-following signal value of “1” as a sell instruction, and on another it ignored both the “five-year-old” and the “ten sentences” constraints, which Agent2 and Agent3 both honoured exactly. No agent fabricated data on the first query; all three fetched the correct values from the DataFrame (Figure 3).

Tab. 1 Backtest comparison of the four trading strategies on Reliance Industries, 2019–2024 (NaN = value not defined in the source)

Metric	Gap Trading	Trend Following	Fibonacci Retracement	Mean Reversion
Annual Return	-1.60%	13.60%	2.50%	3.50%
Cumulative Returns	-7.80%	86.60%	12.90%	18.50%
Annual volatility	7.00%	18.10%	4.20%	7.50%
Sharpe ratio	-0.20	0.79	0.61	0.50
Calmar ratio	-0.10	0.51	2.24	0.37
Stability	0.36	0.42	0.84	0.84
Max drawdown	-15.70%	-26.80%	-1.10%	-9.50%
Omega ratio	0.85	1.21	6.33	1.48
Sortino ratio	-0.29	1.23	3.81	0.78
Skew	2.39	NaN	22.90	0.27
Kurtosis	99.54	NaN	554.41	110.50
Tail ratio	NaN	1.10	NaN	NaN
Daily value at risk	-0.90%	-2.20%	-0.50%	-0.90%

```
# gpt-3.5-turbo and try
agent.run("take all those strategies in account and tell me if I should sell stocks on 29th feb 2024 and why?")

> Entering new AgentExecutor chain...
Thought: I need to find the data for 29th feb 2024 and analyze the strategies to determine if I should sell stocks.
Action: python_repl_ast
Action Input: {"iloc":{"2024-02-29":}}
High                2.595500e+05
Low                 2.875984e+03
Close               2.912680e+03
Adj Close           2.912680e+03
Volume              1.414162e+07
Overnight_returns   -6.445102e-03
Signal_gap_down      6.000000e+00
Close_next_day       2.942926e+03
profit_t1            0.000000e+00
no_t1                2.924315e+03
no_t0                2.772315e+03
Signal_trend_following 1.000000e+00
Close_next_tf        2.942926e+03
profit_tf            6.246990e+01
MACD                 7.305556e+01
Signal_Fibonacci_retracement 0.000000e+00
Close_next_fr        2.942926e+03
profit_fr            0.000000e+00
no_t2                2.835896e+03
std_dev              3.432328e+01
upper_bb             2.997155e+03
lower_bb             2.850557e+03
rsi                  4.278967e+01
Signal_mean_reversion 0.000000e+00
Close_next_mr        2.942926e+03
profit_mr            0.000000e+00
note: 2024-02-29 00:00:00, dtype: float64 note to analyze the data and determine if the strategies indicate that I should sell stocks on this date.
Action: python_repl_ast
Action Input: {'loc': ['2024-02-29']['Signal_gap_down'] and df.loc['2024-02-29']['Signal_trend_following'] and df.loc['2024-02-29']['Signal_Fibonacci_retracement'] and df.loc['2024-02-29']['Signal_mean_reversion']]0.00 indicates that none of the strategies suggest selling stocks on this date.
Final Answer: Based on the data and strategies, it does not seem like you should sell stocks on 29th feb 2024.

> Finished chain.
'Based on the data and strategies, it does not seem like you should sell stocks on 29th feb 2024.'
```

Fig. 2: Example reasoning output of the GPT-powered Agent Executor, which retrieves the strategy signals for a given date and concludes whether to sell (adapted from Tripathi, 2024).

4 DISCUSSION

Why did the larger model read the data better? The likely reason is not finance knowledge but instruction-following. GPT-3.5-Turbo’s two errors were both about following the prompt – treating a numeric flag as a verb, dropping an explicit length limit – rather than about the market. The newer models did not need to “understand trading” to win; they needed to respect the structure of the question and the columns they were given. That points to a practical takeaway for anyone building such a tool: the gain from a better model shows up as fewer misreadings of the user’s instructions, which for a beginner is exactly where damage would happen.

The backtest also warns against reading “best” off a single column. Trend following looks like the winner on return, but its -26.80% drawdown is the kind of swing a new investor rarely sits through. Fibonacci retracement generated a lower absolute return, but its Omega ratio of 6.33 and maximum drawdown of only -1.10% indicate a substantially different risk profile. The strategies should therefore not be compared only by return, because they optimise different combinations of profitability, volatility and down-side risk. A reasonable reading is that a blend (trend following for upside, mean reversion for steady participation, Fibonacci for rare low-risk entries) would give more balanced advice than any one rule, which is the direction the agent’s recommendations leaned.

The main methodological result is that an LLM agent primarily designed for language tasks can provide useful explanations over a clean structured price table. This matters because stock data are already tabular; the contribution of the agent is therefore not price prediction itself, but the conversion of structured signals into understandable investment-oriented explanations. All reported figures come from an in-sample backtest on a single stock without brokerage fees, taxes or slippage, and should therefore be interpreted as optimistic feasibility results rather than as live trading performance. Only proprietary OpenAI models were tried, with no Llama, Mistral

or Falcon, so nothing is known about open-source behaviour. And a backtest is not a live record: slippage and intraday dynamics are absent. Gap trading is the cautionary case. It lost money over the full window yet was profitable for roughly its first three years, which suggests it might still earn its place inside a diversified portfolio rather than being dismissed outright.

5 CONCLUSION

A GPT agent reading a marked-up price table can answer a beginner's investment questions, and the model behind it decides how well. On Reliance data over five years, GPT-4-Turbo handled context and instructions best, GPT-4 stayed accurate to the signals, and GPT-3.5-Turbo made the kind of mistakes that would mislead a novice. Among the strategies the data refuse a single answer: trend following pays the most and risks the most (13.60% annual, -26.80% drawdown), while Fibonacci retracement trades rarely but safely (-1.10% drawdown, Omega 6.33). For a tool aimed at beginners, the results support a design in which a stronger language model explains and compares several strategy signals, rather than presenting any single rule as universally superior. The next steps are the ones this study could not take: open-source models alongside the OpenAI ones, more than one Indian stock, transaction costs and taxes in the backtest, and a path from backtest toward forward testing under real execution.

Acknowledgements

This paper was supported by the project CZ.02.1.01/0.0/0.0/16_017/0002334 Research Infrastructure for Young Scientists, which is co-financed from the Operational Programme Research, Development and Education.

Anthropic Claude was used to summarize the text of the final thesis (Tripathi, 2024) and transform it into the form of this compact paper. The paper was then reviewed and fine-tuned by the author. The author is responsible for the correctness of the final text.

REFERENCES

- AKBAS, F., BOEHMER, E., JIANG, C., KOCH, P. D. 2022. Overnight returns, daytime reversals, and future stock returns. *Journal of Financial Economics*. 145(3), 850–875. <https://doi.org/10.1016/j.jfineco.2021.09.019>
- BANIK, S., SHARMA, N., MANGLA, M., MOHANTY, S. N., SHITHARTH, S. 2022. LSTM based decision support system for swing trading in stock market. *Knowledge-Based Systems*. 239. <https://doi.org/10.1016/j.knosys.2021.107994>
- DEVLIN, J., CHANG, M.-W., LEE, K., TOUTANOVA, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805*. <https://doi.org/10.48550/arXiv.1810.04805>
- EGGEBRECHT, P., LÜTKEBOHMERT, E. 2023. A Deep Trend-Following Trading Strategy for Equity Markets. *The Journal of Financial Data Science*. 5(1), 41–66. <https://doi.org/10.3905/jfds.2023.1.120>
- HO, H.-W., HSIAO, Y.-J., LO, W.-C., YANG, N.-T. 2023. Momentum investing and a tale of intraday and overnight returns: Evidence from Taiwan. *Pacific-Basin Finance Journal*. 82. <https://doi.org/10.1016/j.pacfin.2023.102151>

- KALYAN, K. S. 2024. A survey of GPT-3 family large language models including ChatGPT and GPT-4. *Natural Language Processing Journal*. 6, 100048. <https://doi.org/10.1016/j.nlp.2023.100048>
- LEE, J., STEVENS, N., HAN, S. C., SONG, M. 2024. A Survey of Large Language Models in Finance (FinLLMs). *arXiv:2402.02315* <https://doi.org/10.48550/arXiv.2402.02315>
- LI, Y., WANG, S., DING, H., CHEN, H. 2023. Large Language Models in Finance: A Survey. In: *ICAIF 2023 – 4th ACM International Conference on AI in Finance*, pp. 374–382. <https://doi.org/10.1145/3604237.3626869>
- MACEDO, L. L., GODINHO, P., ALVES, M. J. 2017. Mean-semivariance portfolio optimization with multiobjective evolutionary algorithms and technical analysis rules. *Expert Systems with Applications*. 79, 33–43. <https://doi.org/10.1016/j.eswa.2017.02.033>
- PAN, T., YU, J., ZHENG, L., LI, Y. 2024. Application and Modeling of LLM in Quantitative Trading Using Deep Learning Strategies. *Studies in Computational Intelligence*. 1130, 671–678. https://doi.org/10.1007/978-3-031-50381-8_72
- TRIPATHI, A. 2024. Automated trading system and artificial intelligence. Master's thesis, head: TRENZ, O. Brno: Mendel University in Brno.
- VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., POLOSUKHIN, I. 2023. Attention Is All You Need. *arXiv:1706.03762* <https://doi.org/10.48550/arXiv.1706.03762>
- WANG, L., MA, C., FENG, X., ZHANG, Z., YANG, H., ZHANG, J., CHEN, Z., TANG, J., CHEN, X., LIN, Y., ZHAO, W. X., WEI, Z. and WEN, J. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*. 18. <https://doi.org/10.1007/s11704-024-40231-1>
- ZAKAMULIN, V., GINER, J. 2020. Trend following with momentum versus moving averages: a tale of differences. *Quantitative Finance*. 20(6), 985–1007. <https://doi.org/10.1080/14697688.2020.1716057>

Contact information

Ankit Tripathi: e-mail: qgtripat@node.mendelu.cz

Michaela Veselá: e-mail: michaela.vesela.ui@mendelu.cz

Oldrich Trenz: e-mail: oldrich.trenz@mendelu.cz