European Economic Letters
ISSN 2323-5233
Vol 15, Issue 3 (2025)
http://eelet.org.uk

Adaptive Algorithmic Trading Using Volatility-Guided Reinforcement Learning: Empirical Analysis in Indian Markets

Preeti Gupta[1], Aditya Kasar[2], Mukund Madhav Tripathi[3], Vikrant Arora[4], Sudhanshu Marudgan[5], Anoushka Ramankulath[6]

[1,2,4,5,6] *School of Technology Management & Engineering, NMIMS University, Navi Mumbai, India.*
[3]*School of Commerce, NMIMS University, Navi Mumbai, India.*

Abstract
Equity trading markets often experience periods of price and market volatility. These are driven by various factors, and eventually resulting in significant uncertainty for retail investors and widespread financial losses. Volatile markets often lead to frequent trend reversals, diminishing the effectiveness of traditional signal generation methods. Moving average cross signals, or oscillator signal do not ensure a movement in the indicated direction when raised in a volatile environment. Hence it becomes imperative to explore techniques to navigate volatile markets in an algorithmic framework. This paper proposes a reinforcement learning agent which employs volatility as an input, while trying to make profitable high frequency trades. The strategy was backtested on intraday stock data [750 stocks] from multiple indices listed on the National Stock Exchange of India, representing various sections and sectors of the market. It was found to convincingly outperform the buy and hold strategy in bearish and sideways moving conditions, establishing it as a valid scalping option when trying to diversify and hedge portfolios while employing algorithmic trading. The strategy involves the use of Q-Learning as the major decision-driver, optimized by added stochasticity to respond to volatility.

Keywords: Finance, Reinforcement Learning, Quantitative Analysis, Algorithmic Trading, High Frequency Trading.

1. Introduction
1.1 Algorithmic Trading
An algorithm is a set of instructions used to accomplish a given task. Trading algorithms are computerized models that incorporate the steps required to trade an order in a specific way (Wilhelmina Afua Addy et al., 2024) . This is called automated trading, also known as black-box trading, or algorithmic trading (Borch, 2022).

High Frequency Trading: Using advanced computer algorithms and mathematical models, automated trading allows traders to execute orders in the blink of an eye, taking advantage of small price differences, which is also known as high frequency trading. This rapid approach not only offers better profits potential than traditional trading methods but also attracts a growing number of traders eager to stay ahead in an increasingly fast-paced environment. High frequency trading (HFT) has transformed the landscape of financial markets, accounting for more than 73% of trading volume today (Qin et al., 2023).

1.2 Reinforcement Learning
Reinforcement Learning is a branch of machine learning that utilizes agent- environment frameworks, allowing the agent to learn from its experiences by interacting with the environment via actions and rewards (Ponomarev et al., 2020) . The exploration of actions

allows the agent to autonomously make the best decision for its current state. Reinforcement learning stands out from other machine learning techniques due to its emphasis on exploration, rewards, and learning from both successes and mistakes. Published in 1998, Reinforcement Learning: An Introduction (Sutton & Barto, 2005) , covers the theory, history, and applications of the popular modern RL methods.

Q-***Learning:*** Q-learning is a model-free reinforcement learning algorithm that determines the value of an action in each state. It does not require an environmental model and may deal with environments involving stochastic transitions and rewards, such as financial markets, but offers a simple and computationally inexpensive agent capable of exploring complex environments.

1.3 Indicators

In recent times, algorithmic trading has emerged as a powerful tool for executing trades swiftly and efficiently. Central to the success of algorithmic trading strategies is the utilization of indicators that provide insights into market trends and potential price movements.

Leading Indicators: Leading indicators anticipate price fluctuations. Their major advantage is early signaling for entry and exit, generating additional signals, which allows anticipatory trading practices. They represent price momentum over a specified look-back period. The most prominent leading indicators are the Commodity Channel Index [CCI], Momentum, Relative Strength Index [RSI], Stochastic Oscillator, and Williams %R. Leading indicators can predict extreme price movements well in advance but are also vulnerable to false signal generation.

Relative Strength Index [RSI]: RSI is a momentum oscillator used for analysis of overbought or oversold state of a security. The figure below shows the average RSI construction, which ranges from 0% to 100% (Upreti et al., 2022) . Two horizontal reference lines can be seen: the 30% "oversold" line and the 70% "overbought" line. When the oscillator enters a zone of higher buying pressure compared to recent history, it is referred to as the overbought region (Kaur et al., 2022) . This is frequently a hint that the upward trend is coming to a halt. Similarly, the oversold area denotes the bottom portion of the momentum oscillator where a downswing is ending and there is considerable selling pressure compared to the recent past.

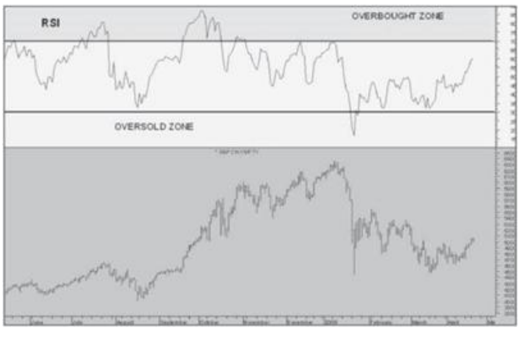

Fig 1: Nifty Chart with RSI (*Technical Analysis Module*, n.d.)

Lagging Indicators: Lagging indicators track trends rather than forecasting them. A lagging indicator displays proof of the occurrence after the occurrence. These indicators function best when prices follow rather extended trends. They do not warn about potential price changes;

instead, they reflect on price movements. Moving averages and Moving Average Convergence/Divergence [MACD] are some well-known examples of lagging indicators. Lagging indicators often reflect price movements with a very observable delay, but they also provide a certain level of robustness to the signals that leading indicators lack.

Simple Moving Average [SMA]: A simple moving average is calculated by taking the mean price of a security over a given number of periods (Upreti et al., 2022), assigning equal value to all prices for the time-period selected. While moving averages can be calculated using the Open, High, and Low-price points as well, the Closing Price is the most used. For example, a 5-day simple moving average is produced by adding the closing prices from the previous 5 days and dividing it by 5.

Crossover Strategies: Crossover strategies are popularly used for automating trading signals when working with moving averages. This strategy involves the use of two moving averages with different periods of calculation. When the shorter period average crosses the longer period average from below, the occurrence is considered a bullish indication and generates a buy signal, and vice versa generates a sell signal. The working logic dictates that the shorter period moving average is more responsive to price movements, and a crossover in any direction is an indication of change in the presiding trend of the instrument.

Exponential Moving Average [EMA]: Exponential moving average is determined by giving more weight to recent values than previous ones, which eliminates latency in basic moving averages. The weighting applied to the most recent price is determined by the moving average's set period. The most recent price receives more weight as the EMA's timeframe shortens (Pardeshi & Kale, 2021). EMAs are used in the calculation of the MACD indicator [Moving Average Convergence and Divergence]. The use of crossover strategies using EMAs amounts to use of the MACD indicator as it uses the same logic to generate buy and sell signals (Kaur et al., 2022). An overbought indication generates a sell signal, and an oversold indication generates a buy signal.

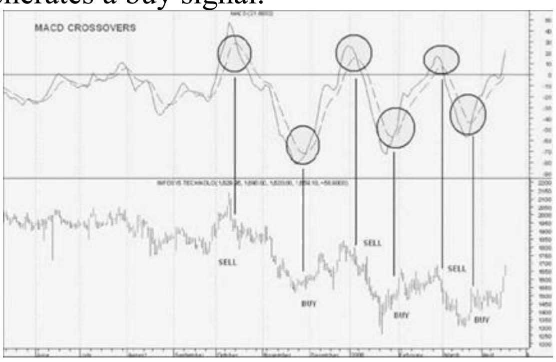

Fig. 2. Infosys chart with crossovers (Technical Analysis Module,)

1.4 Volatility

Markets are considered volatile when price and volume movements cannot be relied on to predict the direction of future movements. It is a feature of a stable and healthy market that a previously established upward or downward trend will continue until external stimuli intervene or momentum naturally dries up. However, volatile markets do not allow established trends to preside over the prevailing market conditions. Volatility is often observed when a primary trend is undergoing reversal, or times of general financial uncertainty, like the periods around general elections or budgets, as well as times of unprecedented liquidity or demand, such as a dividend announcement or the initial public listing of shares of a popular company or commodity. In recent times, it has also been observed that large scale selling or buying in by institutional traders results in spikes in demand and liquidity respectively, resulting in a frenzied volatile market.

It is a characteristic feature of volatile markets that usual technical markers of movements do not hold true. Moving average crosses do not precede any major movements, and oscillators shuffle between overbought and oversold ranges almost spontaneously, without any major price movements following (*Technical Analysis Module*) . This renders typical indicator-based strategies toothless, as they rely on the above-mentioned markers of market movements.

2. Literature Review

Algorithmic trading strategies usually employ a variety of paradigms to automate signal generation. The most common ones include statistical arbitrage, which exploits price inefficiency in valuation, usually via the implementation of forecasting and prediction systems and sentiment analysis, which keeps track of larger public sentiment towards a stock, or a company and makes trading decisions based on the associated positivity or negativity of the sentiment to generate long or short signals respectively.

The two most popular methods of statistical arbitrage employ either indicators to generate trading signals, or machine learning methods to predict expected values of the price, which are then used to preempt corrections and make trades. (Salkar et al., 2021) made use of Relative Strength Index, Moving Average Convergence Divergence and other indicators to automate trading signal generation. In a similar vein, (Paik et al., 2024) employed the Stochastic Oscillator and William's %R indicators to generate trading signals for low frequency trading, tested on American and Korean indices. (Troiano et al., 2018) utilized indicator-based signals and utilized LSTM [Long Short-Term Memory] units to recreate the decisions made based on the indicator values. Among other machine learning methods, (Kamble, 2017) used decision trees to generate trading signals, based on various input variables, including indicator values, dividend announcements, and profit to earnings ratio among others; (Yildirim et al., 2019) proposed an evolutionary optimized system for detection of support and resistance lines, and subsequent strategies. In a novel technique, (Wu, 2021) used Gramian Angular Fields to encode candlestick patterns, allowing the automation of financial technical analysis, and subsequent strategies.

Another popular method of signal generation is via sentiment analysis of texts related to a particular stock or instrument. This requires access to real time information about public sentiment. (Bouktif et al., 2020) uses augmented textual features to predict price movements. In a similar approach, (Patil et al., 2018) used a combined approach including numeric time series to predict price and news analysis to make trading decisions.

Multiple strategies have been proposed which seek to predict or harness market volatility to make trading decisions. (Thavaneswaran et al., 2020) utilizes a novel indicator-based volatility forecasting method to ultimately generate trading signals. (Liang et al., 2020) proposes a dual price and volatility forecasting strategy that in a similar vein is used to evaluate entry and exit points in a dynamically moving market.

Although not very mainstream, the usage of Reinforcement Learning while designing algorithmic trading agents has increased steadily in previous years. The most popular paradigm used is Deep Q-Learning, a variant of Q-Learning which uses a neural network to approximate the Q values instead of a persistent table. This allows the usage of a learnable Q function, which may adapt to various market conditions. (Théate & Ernst, 2021) made use of a DQN [Deep Q-Network] to automate trading, an apt demonstration of Deep Q-Learning. (Y. Li et al., 2019) utilized a DQN with an A3C [Asynchronous Advantage Actor Critic] framework in combination with stacked denoising autoencoders and LSTM units to optimize the agent. Another popular approach is training agents for specific market conditions, and then training a hyper agent to switch between these agents as per the current state, as done by (Qin et al., 2023) and (X. Li & Peng, 2019) , utilizing Deep Q Networks and Markov Decision Processes respectively as sub-agents. Another study that used MDPs to generate trading signals was (X. Li & Peng, 2019) .

Multiple evaluation metrics were used in the related studies, such as Sharpe Ratio, Hit Ratio, and Drawdown, in addition to percentage returns.
As is visible, there are multiple approaches to automate trading, and in an active market environment, multiple such agents might be competing to extract profit, hence the importance of a highly optimized strategy is felt.

3. Proposed Methodology

3.1 Strategy

Like other existing works, the proposed strategy utilizes Q Learning to keep track of previous situations stats and their corresponding Q values. But unlike them, it uses a Q table instead of a network, as the tabular structure is essential to the decision-making process, as will be explained in the next section. The agent allows three possible actions, Buy, Sell, or Hold a particular stock. In this context, the Q-Table serves as a decision matrix, calculating and recording the maximum predicted future profits for each trading action [buy, sell, or hold]. The rows of the Q-Table show various market situations as measured by technical indicators including the Relative Strength Index [RSI], Moving Average Convergence Divergence [MACD], and trade volume, all of them discretized into buckets, each corresponding to a certain mix of these indicators. The columns reflect the agent's market options: purchase, sell, or hold. Each entry in the Q-Table represents the greatest projected future benefit for carrying out a specific action [buy, sell, or hold] given the current state of the indicators [RSI, MACD, volume]. These values are updated iteratively as the agent interacts with the market, allowing the Q-Table to fine-tune its policy related to all possible market states.

The strategy calculates volatility, a sigmoid scaling of the weighted average of price and volume volatility. The calculated value hence ranges between 0 and 1. This value is essential to the decision made by the agent. A volatility of 0 implies that of the three probable decisions, the agent selects the most successful or “appropriate” move for the state, as suggested by the Q table. A volatility of 1 implies that a random decision will be made, utilizing a roulette wheel with all three decisions as sections, and their associated Q values used as probability of selection. A volatility of 0 and 1 implies that the roulette wheel may only contain a subset of decisions available, based on the value. This method allows the agent to respond to a volatile environment by injecting stochasticity into the decision making at times of high volatility and choosing “safer” or “appropriate” moves when the environment is perceived as relatively stable.

3.2 Data

The strategy was intended to be a High Frequency Trading agent, hence requiring data of second or minute granularity. Minute-wise data from the week of 2nd to 6th September was used. As for the selection of stocks, 750 stocks from the 65 equity indices of the National Stock Exchange of India were selected, as illustrated in Figure 3, as of the 10th of September 2024. Data was sourced using the YFinance library on Python, a wrapper for the YahooFinance API.

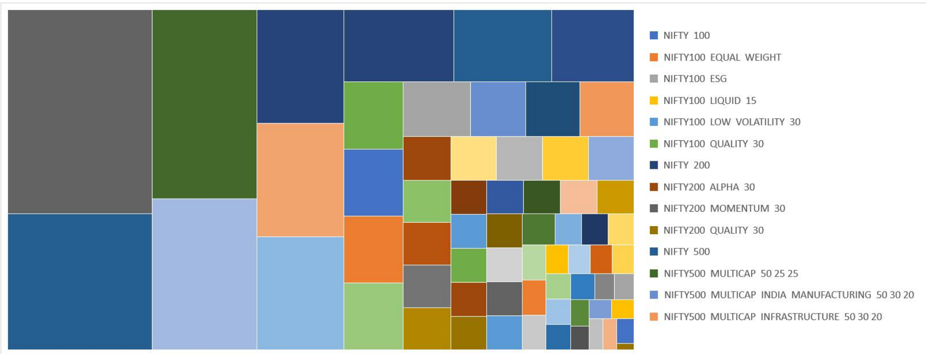

Fig. 3. Stocks per Index, NIFTY

3.3 Experimental Setup and Implementation

The entire implementation of the strategy was done on Python 3.12.1. The Backtrader library was used to simulate real time data retrieval and strategy actions. The strategy itself was natively written in Python, making use of NumPy for mathematical operations. The data for the operations was sourced from the YFinance library as iterated above.

3.4 Evaluation Metrics

The major purpose of this study was to compare the proposed strategy to the buy and hold strategy as a benchmark, and hence percentage returns was deemed sufficient for the evaluation metric.

4. Results and Analysis

Out of the 750 stocks studied, the strategy outperformed buy and hold in 414 stocks. Out of the 414 instances, most stocks experienced bearish or sideways movements over the period of study. Since the strategy inherently exploits stochasticity via roulette wheel selection, the best and worst performances over 30 episodes were kept track of for comparison. Our strategy showed impressive results, outperforming the buy-and-hold approach. This suggests that our method offers a valuable advantage over traditional investment strategies. A sample is shown in Fig. 3.

Stock	Best Strategy PnL%	Worst Strategy PnL%	Buy & Hold PnL%
OIL	0.558000261	-0.531884998	-10.10944246
DHANUKA	0	-0.047195764	-8.558411644
UNOMINDA	1.827687434	0.283407558	-6.200716185
MSTCLTD	0.211490985	-1.464399498	-7.810213038
MASTEK	1.036787412	-0.961284633	-6.686503686
KIRLOSBROS	2.021614101	0.091388896	-5.656196051
NETWEB	0.368056794	-0.943641826	-7.13240248

Fig. 3. A sample of the results

Fig. 4. A sample plot of the strategy's trades

5. Future Scope and Limitations

The results of the strategy can be extracted in different metrics as utilized by other studies for a detailed comparative analysis of performance.

This study only explored the applications of a volatility-based approach in the equity section of financial markets. The strategy could be explored in various environments like other sections such as Forex, Derivatives and Cryptocurrency.

One consideration for future research is the potential for acquiring more extensive data with varied granularity. Access to such data could enable a more in-depth exploration of the strategy's performance, offering richer insights and opportunities for refinement.

6. Conclusion

Volatile markets breed movement reversals and render traditional signal generation obsolete. Moving average cross signals, or oscillator signal do not ensure a movement in the indicated direction when raised in a volatile environment. It hence is of value to explore techniques to navigate volatile markets in an algorithmic framework. While it is evident that volatile markets introduce unexplained randomness to the movements of financial instruments, it is also verifiable demonstratable by the results of the study that stochasticity is a useful tool when utilized with tact. It is a certainty of time, as more financial institutions embrace trade automation, market liquidity and volatility will tread unchartered domains. In such situations, strategies that harness and respond to prevailing market conditions will emerge as more profitable than the rest.

References

1. Borch, C. (2022). Machine learning, knowledge risk, and principal-agent problems in automated trading. *Technology in Society*, *68*, 101852. https://doi.org/10.1016/J.TECHSOC.2021.101852
2. Bouktif, S., Fiaz, A., & Awad, M. (2020). Augmented Textual Features-Based Stock Market Prediction. *IEEE Access*, *8*, 40269–40282. https://doi.org/10.1109/ACCESS.2020.2976725
3. Kamble, R. A. (2017). Short and long term stock trend prediction using decision tree. *Proceedings of the 2017 International Conference on Intelligent Computing and Control Systems, ICICCS 2017*, *2018-January*, 1371–1375. https://doi.org/10.1109/ICCONS.2017.8250694

4. Kaur, A., Bhadauria, M., & Monika, M. (2022). Heuristic approach for forecasting stock price using LSTM and technical indicators. *AIST 2022 - 4th International Conference on Artificial Intelligence and Speech Technology*. https://doi.org/10.1109/AIST55798.2022.10064776
5. Li, X., & Peng, Z. (2019). A novel algorithmic trading approach based on reinforcement learning. *Proceedings - 2019 11th International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2019*, 394–398. https://doi.org/10.1109/ICMTMA.2019.00093
6. Li, Y., Zheng, W., & Zheng, Z. (2019). Deep Robust Reinforcement Learning for Practical Algorithmic Trading. *IEEE Access*, *7*, 108014–108021. https://doi.org/10.1109/ACCESS.2019.2932789
7. Liang, Y., Thavaneswaran, A., & Hoque, M. E. (2020). A Novel Algorithmic Trading Strategy Using Data-Driven Innovation Volatility. *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, 1107–1114. https://doi.org/10.1109/SSCI47803.2020.9308360
8. Paik, C. K., Choi, J., & Vaquero, I. U. (2024). Algorithm-Based Low-Frequency Trading Using a Stochastic Oscillator and William%R: A Case Study on the U.S. and Korean Indices. *Journal of Risk and Financial Management 2024, Vol. 17, Page 92*, *17*(3), 92. https://doi.org/10.3390/JRFM17030092
9. Pardeshi, Y. K., & Kale, P. (2021). Technical Analysis Indicators in Stock Market Using Machine Learning: A Comparative Analysis. *2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT 2021*. https://doi.org/10.1109/ICCCNT51525.2021.9580172
10. Patil, V., Somani, N., Tadvi, A., & Attar, V. (2018). Algorithmic Forex Trading using Combination of Numeric Time Series and News Analysis. *2018 4th International Conference for Convergence in Technology, I2CT 2018*. https://doi.org/10.1109/I2CT42659.2018.9058285
11. Ponomarev, E., Oseledets, I., & Cichocki, A. (2020). Using Reinforcement Learning in the Algorithmic Trading Problem. *Journal of Communications Technology and Electronics*, *64*(12), 1450–1457. https://doi.org/10.1134/S1064226919120131
12. Qin, M., Sun, S., Zhang, W., Xia, H., Wang, X., & An, B. (2023). EarnHFT: Efficient Hierarchical Reinforcement Learning for High Frequency Trading. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(13), 14669–14676. https://doi.org/10.1609/aaai.v38i13.29384
13. Salkar, T., Shinde, A., Tamhankar, N., & Bhagat, N. (2021). Algorithmic Trading using Technical Indicators. *Proceedings - International Conference on Communication, Information and Computing Technology, ICCICT 2021*. https://doi.org/10.1109/ICCICT50803.2021.9510135
14. Sutton, R. S., & Barto, A. G. (2005). Reinforcement Learning: An Introduction. *IEEE Transactions on Neural Networks*, *9*(5), 1054–1054. https://doi.org/10.1109/TNN.1998.712192
15. *Technical Analysis Module*. (n.d.). Retrieved August 13, 2025, from https://www.nseindia.com/learn/self-study-ncfm-modules-intermediate-technical-analysis-module
16. Thavaneswaran, A., Liang, Y., Zhu, Z., & Thulasiram, R. K. (2020). Novel data-driven fuzzy algorithmic volatility forecasting models with applications to algorithmic trading. *IEEE International Conference on Fuzzy Systems*, *2020-July*. https://doi.org/10.1109/FUZZ48607.2020.9177735

17. Théate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications*, *173*. https://doi.org/10.1016/j.eswa.2021.114632
18. Troiano, L., Villa, E. M., & Loia, V. (2018). Replicating a Trading Strategy by Means of LSTM for Financial Industry Applications. *IEEE Transactions on Industrial Informatics*, *14*(7), 3226–3234. https://doi.org/10.1109/TII.2018.2811377
19. Upreti, A., Agrawal, A., Joshi, J. K., & Seniaray, S. (2022). Quantitative study of candlestick pattern & identifying candlestick patterns using deep learning for the Indian stock market. *International Journal of Health Sciences*, *6*(S3), 5739–5749. https://doi.org/10.53730/IJHS.V6NS3.7230
20. Wilhelmina Afua Addy, Adeola Olusola Ajayi-Nifise, Binaebi Gloria Bello, Sunday Tubokirifuruar Tula, Olubusola Odeyemi, & Titilola Falaiye. (2024). Machine learning in financial markets: A critical review of algorithmic trading and risk management. *International Journal of Science and Research Archive*, *11*(1), 1853–1862. https://doi.org/10.30574/IJSRA.2024.11.1.0292
21. Wu, C. (2021). Empowering Financial Technical Analysis using Computer Vision Techniques. *Proceedings - 2021 2nd International Conference on Computer Science and Management Technology, ICCSMT 2021*, 179–184. https://doi.org/10.1109/ICCSMT54525.2021.00043
22. Yildirim, E. O., Ucar, M., & Ozbayoglu, A. M. (2019). Evolutionary Optimized Stock Support-Resistance Line Detection for Algorithmic Trading Systems. *1st International Informatics and Software Engineering Conference: Innovative Technologies for Digital Transformation, IISEC 2019 - Proceedings*. https://doi.org/10.1109/UBMYK48245.2019.8965471