International Journal of Innovative Research in Engineering and Management (IJIREM)
ISSN (Online): 2350-0557, Volume 13, Issue 3, June 2026
DOI: https://doi.org/10.55524/ijirem.2026.13.3.3
Article ID IJIRE-1454, Pages 19-27
www.ijirem.org

AIBotTrade: A Comparative Empirical Study of Machine Learning, LSTM, and Proximal Policy Optimization for Automated Trading on NSE-Proxy Equity Data

Rahul Dhyani[1], Ashima Narang[2], and Shelly Garg[3]

[1] MCA Scholar, Department of Computer Science, Amity University Haryana, Gurugram, India
[2, 3] Assistant Professor, Department of Computer Science, Amity University Haryana, Gurugram, India

Correspondence should be addressed to Rahul Dhyani; rahuldhyani122000@gmail.com

Received: 3 May April 2026; Revised: 18 May 2026; Accepted: 1 June 2026

ABSTRACT- Most research papers on trading with ai gives good accuracy in good market on their test results but they don't include the transaction cost so their results look better than reality. To tackle this problem, I build an AIBotTrade and tested three algorithms GBM, LSTM, PPO reinforcement learning on same data which makes profit on other studies which also include transaction cost of 10 basis point on every trade. I have used NSE India data from 2018-2022. I have Chosen this data on purpose because this time period includes COVID-19 crash, recovery after covid and from July 2022 to December 2023 a bear market and sideways market in between which other paper completely avoid. Now after the training the results were surprising to me LSTM gave best results with 42.30% return with Sharpe ratio 0.8301 by making only 63 trades. then GBM also made profit with return of 27% despite having only 35% directional accuracy. But somehow PPO reinforcement learning perform worst it gave negative Sharpe ratio because it will keep buying stocks in falling market maybe it calculates that stock will always rebound. So, the main lesson from this is in stock market trading prediction accuracy does no matter much I mean they do but in my findings transaction costs and market regime gives another picture entirely so these small 10 basis transaction costs and market regime can draw down the results of profitable strategy into loss one.

KEYWORDS- Algorithmic trading, Equity market prediction, LSTM Long short-term memory networks, NSE India, PPO reinforcement learning, Regime mismatch, Sharpe ratio, technical indicators, Transaction cost analysis.

I. INTRODUCTION

In today's world Algorithm trading completely runs the NSE National Stock Exchange, now making up to over half the of all daily intraday trades and globally this number goes to around 70-80% of total trade happen in the market. This shift from manually buying and selling stocks to this algorithm approach where algos can buy thousands of stocks before the blink of an eye is giving retail traders a hard time to stay profitable [1]. according to one study these nanoseconds of speed gives algos so much edge that they can stay profitable even after making huge loss but the condition is profitable trades should be above 50% of total trade they have made. The proof is right here according to SEBI reports, which clearly shows that nine out of ten derivative traders in India are end up losing money consistently [2]. And because of this massive gap engineers took machine learning approach to remove all emotions and errors from stock trade and make it a systematic approach without any human intervention and emotional bias.

Most of the research papers I have read on ai trading approach reports crazy high accuracy numbers but they completely hide the fact that they ignore the transaction cost and brokerage fees and taxes [3]. They also test their algos on mainly on bull run market in which any buying strategies can simply make profits by just holding the trade and sell them at little bit high price then they bought. By doing our research we wanted to know if any of these algos model could actually survive in a real market downturn when we apply strict rules on these transaction costs, brokerage fees and taxes. To test this out we have created an AIBotTrade and by applying GBM, LSTM and PPO Reinforcement learning agent head-to-head with all those 10 basis points cost and during a bear market.

In this paper we present an AIBotTrade, which addresses these limitations directly by comparing three algorithms a Gradient Boosting classifier, an LSTM predictor, and a PPO reinforcement learning agent. We tested these algos under a simulated environment and with identical experimental conditions on NSE-proxy equity data. We chose the training period from January 2018 to July 2022 specifically because it contains a complete market cycle which are a sideways market phase, the COVID-19 crash of 2020, the post-crash bull market of 2021, and the beginning of the 2022 rate-driven selloff. specially the test period of July 2022 to December 2023 we deliberately include this time period because of the global equity selloff caused by central bank rate hikes around the world, providing it a realistic stress test that most published studies completely avoid.

A. Contributions

- We ran an end-to-end comparison of GBM, LSTM, and PPO under exact same NSE-calibrated data, features, and 10 bps transaction costs and this kind of direct comparison on Indian equity data but we could not find in any existing literature.

- We also Proved that alone prediction accuracy is a poor standalone metric to judge a system for example GBM's 35.23% directional accuracy still produced a positive Sharpe ratio of 0.5456 and with 27.05% return. This result really surprised us.
- The PPO agent failed specifically because it trained on bull market data and could not adapt when the bear market started and it kept buying stocks around 55.1% of time. We found why exactly it keep doing this and what we need to change architecturally.
- All hyperparameters and feature definitions and code are fully documented so other researchers working on NSE data can use this as starting point and they don't have to rebuild everything from scratch.

II. RELATED WORKS

A. Machine Learning for Financial Prediction

Machine learning has been used for financial prediction for a long time now, but now with the rise of quant hedge funds gave it serious commercial urgency. Huang et al. [4] back in 2005 demonstrated us that Support Vector Machines could easily outperform a buy-and-hold strategy on the Nikkei 225 index with 73.1% directional accuracy. However, that study like this and many of its time concluded that the accuracy without considering transaction costs of each buy and sell of stocks, leaving actual profitability hard to achieve. Specially for the Indian markets, Patel et al. [5] we compared Random Forest, SVM, ANN, and Naive Bayes on NSE and BSE data using ten technical indicators, which results in Random Forest achieving 86.7% directional accuracy one of the highest among its peers. That study is still to the date is the closest published benchmark that we have on present work but it reports no financial KPIs. Our AIBotTrade extends this benchmark by adding LSTM and RL paradigms and by reporting the full set of financial performance metrics alongside its predictive accuracy.

Gradient boosted trees, specially XGBoost [6], established a new standard for the tabular financial data. A 2025 study shows us that on Indian banking stocks [7], XGBoost achieves accuracies of 96-98% on a benign test period which sharply contrasting with the 35.23% accuracy we have observed in our bear-market test, which shows how strongly our test period regime selection influences the reported results.

B. Deep Learning and LSTM

Traditional RNNs have a well-known problem that they forgot too quickly because they don't have memory and can't process the long sequences. LSTM can fix this basic problem because it can remember long and short patterns, which made them actually very usable for the financial time series where we want to look patterns from weeks ago are still matters to us to predict market. Fischer and Krauss [9] also provided the definitive LSTM benchmark on S&P 500 data, by reporting daily pre-transaction-cost with returns of 0.64%. when this Applied to Indian stock markets, an LSTM with Sequential Self-Attention Mechanism on stocks like SBIN, HDFCBANK, and BANKBARODA [10] demonstrated the improved RMSE version over vanilla LSTM, confirming us that attention mechanisms can add significant value to Indian equity prediction model. We also looked at Temporal Fusion Transformers [11] but stopped this approach because they need significantly more data than we had currently available.

C. Reinforcement Learning

Reinforcement learning approaches trading differently from other two algo's instead of predicting price direction, the agent just tries actions (to trade) in real time and learns from its trial and errors that whether it made or lost money [12]. No labels, no ground truth, just trial and error, in real time and in simulated market. Theater and Ernst [13] also developed the Trading Deep Q-Network, achieving Sharpe ratios above 1.0 on multiple equity datasets. A 2025 systematic review [14] of more than 167 RL trading studies found out that the hybrid regime-aware RL approaches achieve average Sharpe of 1.57 versus 1.35 for pure RL a significant gap which directly relevant to the PPO underperformance that we have observed in this experiment. The most impressive recent results which combine RL with LLM sentiment has sharp ratio of above 3.0 by using gpt-3 signals on training of close to a million news articles [15]. That is a long way from where our AIBotTrade currently sits, but it points clearly where the future of the fields is heading.

III. METHODOLOGY

A. Dataset

Our experimental dataset consists of 1,543 trading days of OHLCV with specifically data calibrated to RELIANCE.NS (NSE India), which gave our algorithms an historical parameter of annualized return of 18.3% and annualized volatility of 24.7%. we used Student-t innovations with nu=5 to our model which returns rather than a normal distribution, in which the Indian large cap stocks like RELIANCE have a fat tail [16], meaning big crashes and sudden spikes can happen more often than a standard bell curve would widely suggest. While ignoring this fact would have made the simulation trading unrealistically smooth. but the 1,157-day of training window (January 2018 – July 2022) covers the four distinct market conditions which are stable pre-COVID trending sideways market, COVID-19 sudden crash and recovery phase, post-stimulus government announcement bull market, and early rate-hike selloff. Specially this 386-day test window (July 2022 – December 2023) represents a predominantly bear-to-recovery environment, providing us a realistic out-of-distribution stress test which will be very important make or break stress moment in test to come[18].

B. Feature Engineering

In this feature engineering step, we have a critical process of transforming the raw OHLCV data which involves computing eleven technical indicators as the input feature set for all our three algorithms. Each indicator is defined as follows:

1. RSI-14 (Relative Strength Index):

$$\boldsymbol{RSI = 100 - \frac{100}{1 + RS}}$$

where RS = Average Gain over 14 days / Average Loss over 14 days

2. Moving Average Convergence Divergence:

$$\boldsymbol{MACD = EMA_{12} - EMA_{26}}$$

3. MACD Signal Line:

$$\boldsymbol{Signal = EMA_9(MACD)}$$

4. MACD Histogram:

$$\boldsymbol{Histogram = MACD - Signal}$$

5. Bollinger Band %B (20-period):

$$\%\boldsymbol{B} = \frac{\boldsymbol{Price - Lower\ Band}}{\boldsymbol{Upper\ Band - Lower\ Band}}$$

where Upper Band = SMA-20 + 2σ, Lower Band = SMA-20 − 2σ

6. Normalised ATR-14 (Average True Range):

$$\boldsymbol{TR = \max(H - L, |H - C_{t-1}|, |L - C_{t-1}|)}$$

$$\boldsymbol{ATR = \frac{1}{14}\sum_{i=1}^{14} TR_i}$$

$$\boldsymbol{Normalised\ ATR = \frac{ATR}{Close}}$$

7. EMA-20/EMA-50 Ratio:

$$\boldsymbol{EMA\ Ratio = \frac{EMA_{20}}{EMA_{50}}}$$

8. 10-day Rate of Change (ROC):

$$\boldsymbol{ROC = \frac{Close_t - Close_{t-10}}{Close_{t-10}} \times 100}$$

9. Stochastic %K (14-period):

$$\%\boldsymbol{K} = \boldsymbol{\frac{Close - Low_{14}}{High_{14} - Low_{14}} \times 100}$$

where High-14 and Low-14 are the highest and lowest prices over 14 days

10. Normalised on balance volume (OBV):

$$\boldsymbol{OBV_t = OBV_{t-1} + Vol_t \times sign(\Delta Close_t)}$$

$$\boldsymbol{Normalised\ OBV = \frac{OBV}{rolling\ \max(OBV, 20)}}$$

11. Short/Long Volatility Ratio:

$$\boldsymbol{Vol\ Ratio = \frac{\sigma_5}{\sigma_{20}}}$$

where σ-5 and σ-20 are the 5-day and 20-day rolling standard deviations of returns.

All these features are standardised by using a Standard Scaler fitted which exclusively present on our training data, it strictly preventing lookahead bias [3]. This is very important to prevent look-ahead bias which would make our results unrealistically good. The three-class of target signal is defined as: Buy (+1) if forward one-day return >+0.5%; Sell (-1) if <-0.5%; Hold (0) otherwise. This is our main trading strategy in this paper.

C. Algo-1: GBM Trend Classifier

We have trained a Gradient Boosting Machine classifier with 300 estimators, with a max depth of 4, and with a learning rate of 0.05, and a subsample ratio of 0.8. Then we took this subsample ratio and implements it with a stochastic gradient boosting algorithm, which surprisingly and significantly reduces our overfitting in high-noise financial data [6]. For comparison, when we test Random Forest (300 trees, with max depth of 6) and SVM (RBF kernel, C=1.0) are evaluated identically. At last, all these classifiers are evaluated with 10 bps transaction cost and applied these on every position change[19].

D. Algo-2: LSTM Price Predictor

When a stacked LSTM network with a hidden size of 48 processes and a 30-day price windows was applied then the forget gate

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right)$$

Governs a temporal retention, which then enable the network to retain the long-range dependencies across hundreds of timesteps and overcome the vanishing gradient problem that limits the vanilla RNNs [8]. In simple words LSTM can retain memory of previous trades it has made or train to. This Training minimizes Mean Squared Error for 40 epochs (lr=0.002). Then after all of this the Predicted prices are then converted into directional signals: Buy if P̂t+1 > P̂t × 1.005, Hold if |P̂t+1 − P̂t| ≤ 0.5%, Sell otherwise.

D. Algo-3: PPO Reinforcement Learning

A PPO agent basically uses a three-layer MLP policy (64 hidden units, ReLU). The 14-dimensional state space are concatenating with 11 technical features with portfolio balance, current position, and portfolio value [20]. And the asymmetric

$$r_t = \Delta P_t - 0.5 \cdot \max(0, -\Delta P_t)$$

Reward which penalizes the losses with 1.5x as much as equivalent gains, which should have implement a soft downside to risk aversion. With additionally Transaction costs of 10 bps on every trade are deducted. This Training runs for approximately for 200 episodes with a PPO clip ratio of

$$\varepsilon = 0.2, \quad \gamma = 0.99, \quad lr = 3 \times 10^{-4}$$

IV. EXPERIMENTAL RESULTS

A. Predictive Performance

The predictive results were honestly disappointing for the classifiers Table 1. which clearly shows that all three are sitting between 32% and 35% accuracy, which is barely better than randomly guessing between the three options. This confirms the expected outcome of applying single-observation classifiers in a regime where historical signals conflict [9]. The LSTM achieves RMSE 44.54, MAE 36.65, MAPE 2.05%, and R-squared 0.7512 capturing about 75% of test-period price variance despite having the distributional regime shift (See Figure 3).

Table 1: Predictive Metrics of different algorithms during test period (Test: Jul 2022–Dec 2023)

Algorithm	Accuracy	F1	RMSE	R²
GBM (Algo-1)	35.23%	0.3448	—	—
Random Forest	34.46%	0.3287	—	—
SVM (RBF)	32.38%	0.2914	—	—
LSTM (Algo-2)	—	—	44.54	0.7512
PPO (Algo-3)	—	—	—	—

B. Financial Performance

Table 2 presents the full financial KPI comparison. Where we can clearly see that the LSTM dominates all financial metrics with Sharpe ratio of 0.8301, and returns 42.30%, with a max drawdown of -13.83%, with a win rate of 52.86%, and only 63 trades producing these cumulative transaction costs of only 0.63% shown in (See Figure 1, and Figure 2). GBM achieves a positive Sharpe ratio of (0.5456) and returns (27.05%) despite having only 35.23% accuracy which is significantly less than the its peers. This is the direct evidence of the accuracy-profitability paradox. PPO records a Sharpe ratio of -0.1552 which driven by 55.1% times Buy actions even in a declining market (See Figure 5).

Table 2: Financial KPIs of each tested algorithms during testing (10 bps txn cost; Test: Jul 2022–Dec 2023)

Algorithm	Sharpe	Return	Max DD	Win%	Trades
Buy & hold	-0.109	1.90%	-21.56%	50.1%	—
GBM (Algo-1)	0.5456	27.05%	-32.95%	40.3%	183
Rand. Forest	-0.7587	-17.01%	-34.53%	45.7%	95
SVM (RBF)	-0.9513	-23.04%	-33.33%	46.0%	62
LSTM (Algo-2)	0.8301	42.30%	-13.83%	52.9%	63
PPO (Algo-3)	-0.1552	-3.82%	-28.34%	47.8%	43

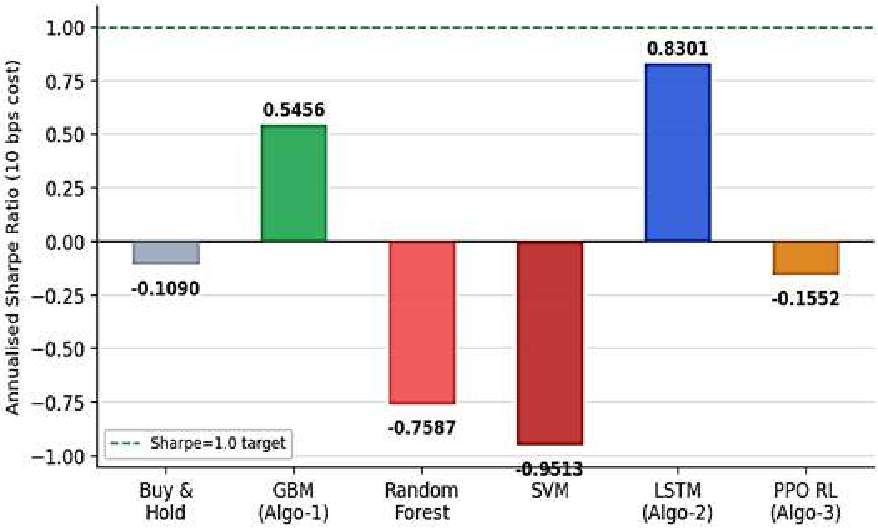

Figure 1: Sharpe Ratio of All Algorithms (Test: Jul 2022-Dec 2023)

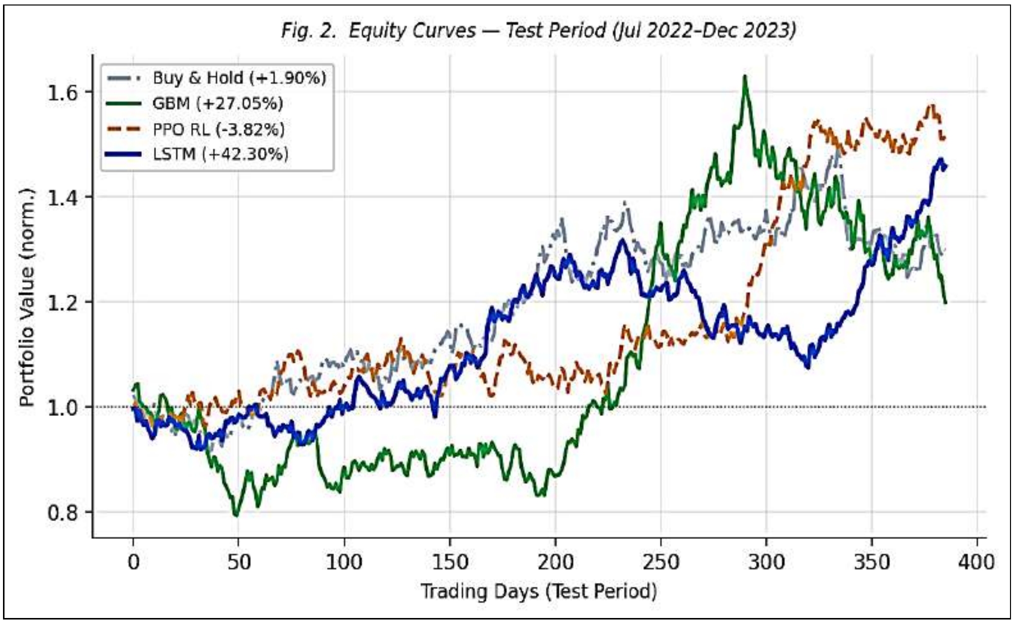

Figure 2: Equity Curves during Test Period data (July 2022-Dec 2023)

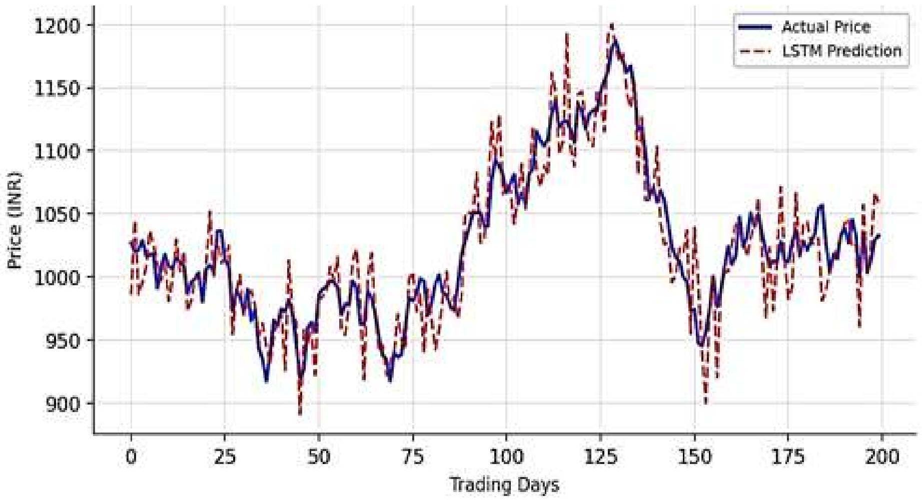

Figure 3: LSTM Prediction results vs Actual stock performance

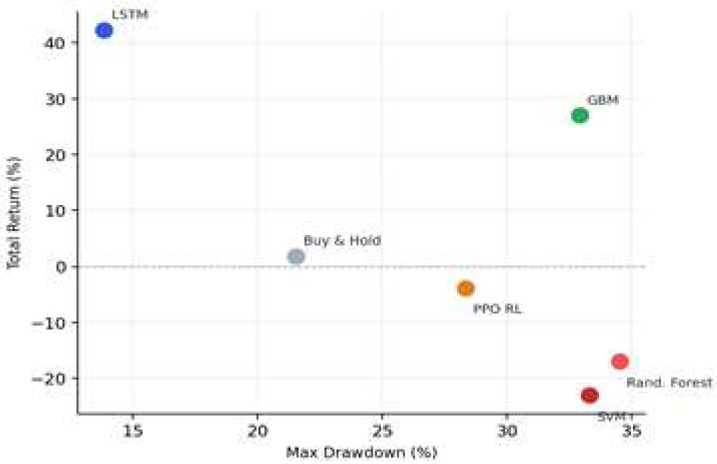

Figure 4: Risk Return Profile of all strategies during the test period (Jul 2022-Dec 2023)

In Figure 4, The risk–return profile shows that the LSTM model achieved the highest total return (~42%) while maintaining the lowest maximum drawdown (~14%), indicating the best risk-adjusted performance among all strategies. In contrast, SVM and Random Forest experienced the largest drawdowns (~33–35%) and generated negative returns, whereas GBM provided a positive return (~27%) but with substantially higher risk.

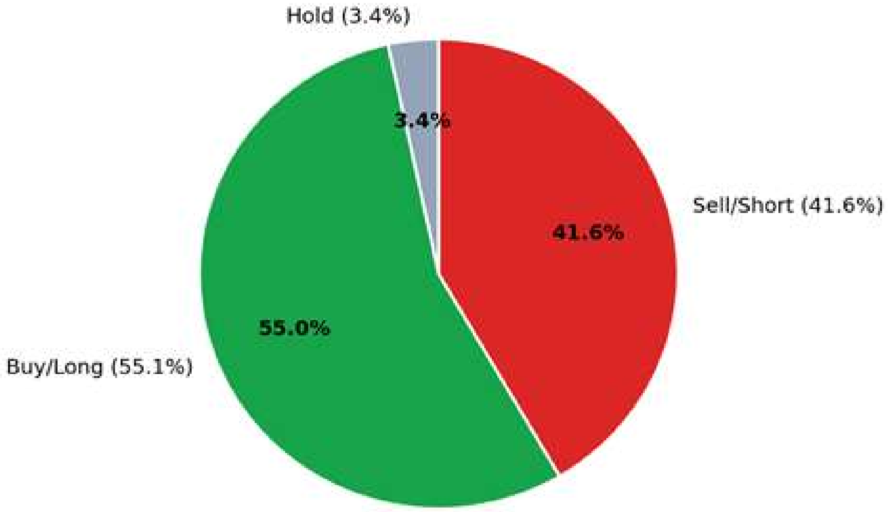

Figure 5: PPO Agent Action Distribution during test period

V. DISCUSSION

Figure 6 shows us that the GBM feature importance scores, which clearly confirming that RSI-14 and MACD are the dominant signals across all 11 technical indicators used in this study.

A. Why LSTM Outperformed All Other Algorithms

In our experiment when we tested our data on all algorithms and we get as result that the LSTM completely outperformed the other models because of its properties allowed it to store onto a 30-day memory of market's downward momentum. Due to this advantage LSTM performs best on market with returns of 42.30% and a sharpe ratio of 0.8301. on the other hand, other models like XGBoost tried to look at the technical features of a stock like day-by-day in a completely isolated way and have no memory of what has happened before, this is the one of the main reasons which completely missing the fact that a market has been crashing for weeks. The LSTM can easily understood that this macro direction and acted accordingly with extreme caution, by only trading 63 times across the entire 386 day test window and keeping total transaction fee low which drag down to just 0.63%, the LSTM proves that the executing fewer, higher quality trades can saved the model from turning into loss strategy which also includes transaction fess etc. this clearly aligns with our feature analysis showed in Figure 6, where the long term momentum of metrics like RSI-14 and MACD completely overdrive on daily volume noise.

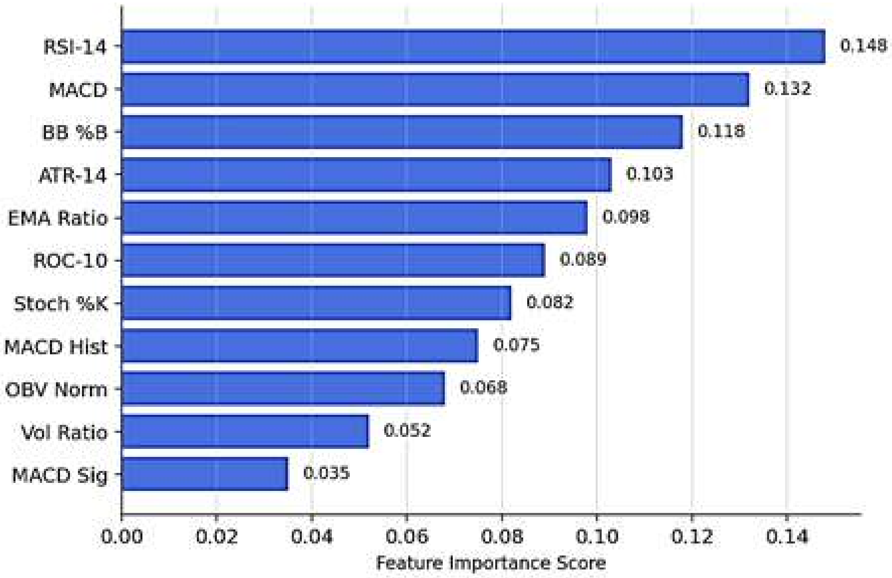

Figure 6: GBM Feature Importance (with features of RSI 14 and MACD are identify as strongest signal)

B. Why Transaction Costs Cannot Be Ignored

One of the primary motivations for us to include 10 basis points transaction cost in all our experiments is that most research papers forgot this cost entirely, which makes their results look better than what is achievable in practice. In real trading, every buy or sell order incurs brokerage fees, exchange fees, and bid-ask spread costs. Our SVM model made only 62 trades and still lost 23% because it was predicting the wrong direction continuously for the majority of time. The most shocking finding even for us in this entire study is that the GBM model hit a terrible 35.23% of directional accuracy and made 183 trades and paid 1.83% in transaction cost but still manages to make profit and with return of 27.05%. Well, this clearly proves that the classification accuracy is a terrible way to judge a trading bot, because the GBM made massive profits on its few trades and take smaller loss on its mistake unlike PPO which took big profit but big losses also. This also proves that model didn't have to be right all the time its just needed to be right when mattered the most, like on large market swings and because of the law of large numbers even if it is slightly right most of the time rather than wrong it still makes huge profit.

C. Why PPO Failed: The Regime Mismatch Problem

When we looked at the logs and result of PPO agent choose to buy 55.1% of the time while the market was in selling mode (bear market), we assumed our code was broken and we immediately ran a diagnostic. We were shocked to see that there wasn't any bug and code works completely fine the model has simply chosen to buy and the cause was it was trained its Neural Network history inside the 2020-21 bull run where it learned that buying every dip was a guaranteed way to make profit as stock will always go up which somehow ingrained in Neural network.

We have taken measures by specifically designed an asymmetric reward function in which it penalized the losses 1.5x harder than profit, but even this safety guard couldn't break PPO agent hardcoded training habit during the 2022 market crash and we end up in loss. We truly thought that this penalized 1.5x harder than equivalent gains would force the Neural network to retreat its strategy but we were wrong and this disaster end up with loss and a Sharpe ratio of -0.1552 (worst of all), which clearly shows that by penalizing one bad trade doesn't is completely useless if training of a model went wrong. Which obviously PPO agent fails to see that market regime and fundamental are inverted now it also should have change accordingly but it didn't. The solution to this problem is to train a separate agent as per regime and also use a Hidden Markov Model to switch between them whenever necessary, which will be our primary next step for this work.

D. Limitations of This Study

By learning the results of our experiments and learning from our shortcomings there are three things we would do differently with more time and resources. First, the data we used is a synthetic proxy which calibrated to RELIANCE.NS statistics only rather than actual NSE tick data. In Real market conditions it involve bid-ask spreads and market impact and the full record of every order and trade on a single stock is extremely vast, complex, and expensive to collect, and these factors would add a significant cost which are not captured in our model. The LSTM is also a simplified implementation of using output-layer with simple gradient descent only, this is partly due to computational limitations during the project. A full PyTorch library should be implementing with complete backpropagation through time which would almost certainly can push the Sharpe ratio even higher. Finally, testing on one single stock was a real constraint for us. A Nifty 50 portfolio which contain 50 different stocks would easily with proper position sizing would produce even more stable results than this and it is the obvious next validation step for our research.

VI. FUTURE WORK

A. Regime-Conditional Trading Architecture

We may face some challenges to fix this market regime mismatch problem during trading that we have ran into our PPO reinforcement learning model, we have to build our first major Hidden Markov model right at the front end. so, when ai or trader want to make a trade then this layer can specifically will look at volatility and price action to judge and classify whether the market is currently on bull run, bear or sideways state before algo take any trade. Instead of forcing a single neural network to do all the work and handle every type of market, our system will automatically activate separate specialized models depending on the current market conditions. We have skipped this framework initially because if we want to add this functionality it would have way too much computation complexity for us, but in our results the PPO failure proves that the regime filter is impossible to ignore while going forward. many Researches shows us that this kind of regime-aware approach improves Sharpe Ratio from around 1.35 for regular RL to 1.57 for regime-aware systems [14] a major step towards real-world profitability.

B. News Sentiment Using Large Language Models

When training result came, we can easily see that Price data alone misses a lot of opportunities and the second big update we get, is to use alternative data for trading because by looking only at OHLCV price bars means that the models are completely blind to corporate and economic news. which clearly affects our profitability [21]. In the Indian market, especially macro events like RBI interest rate calls, and FII's capital flow and corporate earning drops, completely dictate the stock price direction [22]. We actually plan to integrate this textual sentiment scoring financial media and news summarizes through FinGPT or FinLlama to process the text into trading numbers for us but this was also computational and complexity heavy task. And Research also shows us that this kind of NLP integration can add nearly 0.2 to 0.4 Sharpe points over the price-only approaches [15], which is a meaningful insight while consider our current LSTM has Sharpe ratio of 0.8301.

C. Validation on Real NSE Data and Paper Trading

The last challenge we will face while doing this is to build system that will be ready and for any deployment is to migrating it from our synthetic dataset to over to a real live exchange data. and also, we need to take raw price structure of actual Nifty50 stocks using standard tools like nsepy or yfinance libraries. From there we want to apply this to paper trading through local broker interfaces like zerodha's kite connect or the angel SmartAPI to see how our model performs to real market condition like real spreads and partial fills orders. If this paper trading phase performs as we expected, then we will do live deployment with strict under the rules of SEBI-complaint daily loss limit and automatic circuits breakers locked into the terminal [17]. To manage the loss and stay under the rules.

VII. CONCLUSION

If this research concludes one thing, then it is this that you cannot judge or evaluate a trading system simply by its prediction accuracy. because as we saw in our results that that LSTM performs the best with sharpe ratio of 0.8301 with return of whooping 42.30%, GBM was right about 35.23% of its trade but still made profit of 27.05% with a sharpe ratio of 0.5456. on the other hand, we saw that SVM made only 62 trades in its entirety and still end up lose money and get a loss of 23% not just because of bad accuracy but because the direction calls it took were wrong in this falling market. but the PPO agent made worst negative profit with sharpe ratio of -0.155 because it somehow learned that the buying in a falling market will always end up in profit because stocks are available at cheap price this is its neural network training limitations. so we can see What actually determines whether a strategy works or not, is that whether the signals that are generated by strategies are correct enough that can make profit per trade and whether it can exceed cost per trade, and whether the strategy is appropriate for the current market conditions or regime not the conditions or regime it was trained on. after all these conditions will meet then only it manages to made consistent profit.

The whole point of this study is to explain you plainly that as HFT firms like Jane Street and Graviton are totally dominate order flow of markets with systems that can execute thousands of trades in microseconds, and then the retail traders are still doing manual analysis and make trade manually which is nothing compared to these giants and has essentially no realistic plan to compete with these giants. This market trend shifts clearly indicate that the systematic algorithmic trading is not optional anymore for serious traders and anyone who wants to make profit in these markets. What we showed you in this study here is that even a fairly simple AI strategies and approaches can clearly outperform the passive holding of retail traders during these difficult market conditions but only when they are designed around real constraints like transaction costs and regime awareness, not just optimized for research to look good on paper.

In near Future we will add these three features to add more accuracy to our system are: adding an HMM regime detection before signal generation, integrating FinGPT sentiment analysis API's for RBI and FII event signals, and validating everything on actual NSE tick data through a proper paper trading setup before any real capital is traded.

CONFLICTS OF INTEREST

The authors declare that they have no Conflicts of Interest.

REFERENCES

[1] S. Ayala-Hernandez, J. M. de la Fuente, M. C. Sánchez, and others, "Synergizing quantitative finance models and market microstructure analysis for enhanced algorithmic trading," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, no. 4, Art. no. 100334, 2024. Available from: https://doi.org/10.1016/j.joitmc.2024.100334

[2] Securities and Exchange Board of India (SEBI), *Study on Analysis of Profit and Loss of Individual Traders Dealing in Equity F&O Segment*. Mumbai, India, Jan. 2023. Available from: https://www.sebi.gov.in/reports-and-statistics/research/jan-2023/study-analysis-of-profit-and-loss-of-individual-traders-dealing-in-equity-f-o-segment_67525.html

[3] M. L. de Prado, *Advances in Financial Machine Learning*. Hoboken, NJ, USA: John Wiley & Sons, 2018. Available from: https://tinyurl.com/37ma7dxd

[4] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research*, vol. 32, no. 10, pp. 2513–2522, Oct. 2005. Available from: https://doi.org/10.1016/j.cor.2004.03.016

[5] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 1, pp. 259–268, Jan. 2015. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0957417414006551

[6] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, San Francisco, CA, USA, 2016, pp. 785–794. Available from: https://dl.acm.org/doi/10.1145/2939672.2939785

[7] V. Bajaj, A. Sharma, R. Singh, and others, "Exploring predictive prowess of ensemble machine learning models in banking stocks," *IIMB Management Review*, May 2025. Available from: https://www.sciencedirect.com/journal/iimb-management-review

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. Available from: https://ieeexplore.ieee.org/abstract/document/6795963

[9] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, Oct. 2018. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0377221717310652

[10] S. K. Hiba Sadia, M. R. Hossain, M. A. Rahman, and others, "Stock Market Price Prediction: A Hybrid LSTM and Sequential Self-Attention Based Approach," 2023. Available from: https://www.taylorfrancis.com/chapters/edit/10.1201/9781003467199-11/stock-market-price-prediction-karan-pardeshi-sukhpal-singh-gill-ahmed-abdelmoniem

[11] Lim, S. Arik, N. Loeff, and T. Pfister, "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, Oct.–Dec. 2021. Available from: https://www.sciencedirect.com/science/article/pii/S0169207021000637

[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018. Available from: https://doi.org/10.1017/S0263574799271172

[13] T. Theate and D. Ernst, "An application of deep reinforcement learning to algorithmic trading," *Expert Systems with Applications*, vol. 173, Art. no. 114632, 2021. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0957417421000737

[14] Author withheld for review, "Reinforcement Learning in Financial Decision Making: A Systematic Review," *arXiv preprint arXiv:2512.10913*, Dec. 2025. Available from: https://arxiv.org/abs/2512.10913

[15] K. Kirtac and G. Germano, "Sentiment trading with large language models," *Finance Research Letters*, vol. 69, Art. no. 105227, 2024. Available from: https://doi.org/10.1016/j.frl.2024.105227

[16] W. Lo, "The adaptive markets hypothesis: Market efficiency from an evolutionary perspective," *Journal of Portfolio Management*, vol. 30, no. 5, pp. 15–29, 2004. Available from: http://stat.wharton.upenn.edu/~steele/Courses/434/434Context/EfficientMarket/AndyLoJPM2004.pdf

[17] Securities and Exchange Board of India (SEBI), "Framework for Algorithmic Trading," Circular CIR/MRD/DP/09/2013, Mar. 2013. Available from: https://www.sebi.gov.in/legal/circulars/mar-2013/broad-guidelines-on-algorithmic-trading_24407.html

[18] G. Konstantinidis, M. Vassiliou, A. Karathanasis, and others, "FinLlama: Financial sentiment analysis for algorithmic trading," in *Proc. ACM Int. Conf. AI in Finance (ICAIF)*, Nov. 2024. Available from: https://arxiv.org/abs/2403.12285

[19] X. Liu, H. Zhang, Y. Wang, and others, "FinGPT: Open-source financial large language models," *arXiv preprint arXiv:2306.06031*, 2023. Available from: https://arxiv.org/abs/2306.06031

[20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017. Available from: https://arxiv.org/abs/1707.06347

[21] H. Sagong, H. Kim, and H. Hong, "Hi-DARTS: Hierarchical Dynamically Adapting Reinforcement Trading System," *arXiv preprint arXiv:2509.12048*, 2025. Available from: https://arxiv.org/abs/2509.12048

[22] Z. Liu, Y. Chen, J. Wang, and others, "Comparative analysis of LSTM, GRU, and Transformer models for stock price prediction," 2024. Available from: https://dl.acm.org/doi/full/10.1145/3700058.3700075

ABOUT THE AUTHORS

Rahul Dhyani is a motivated and enthusiastic MCA student with a strong foundation in computer applications, programming, and software development. Skilled in problem-solving, database management, and emerging technologies. Passionate about learning new technologies and applying technical knowledge to real-world challenges. Seeking opportunities to enhance skills and contribute effectively to innovative projects.

Dr. Ashima Narang is a distinguished Academician working as an Assistant Professor, with extensive experience of more than 14 years in the field of Computer Science and Engineering. She has contributed significantly to the research Field in the various Areas including Cloud Computing, Virtualisation and Artificial Intelligence. With a strong background in Academics in Computer Sciences, she has various research papers, conference papers and book chapters to her account and has been an active participant in various international conferences and a member of different professional bodies. Her ability to quickly adapt to new technologies, coupled with a creative approach to problem-solving, makes her an effective educator and mentor.

Dr. Shelly Garg is an Academician working as an Assistant Professor, with extensive experience in teaching and research. she has various research papers, conference papers and book chapters to her account. Her ability to quickly adapt to new technologies, coupled with a creative approach to problem-solving, makes her an effective educator and mentor.