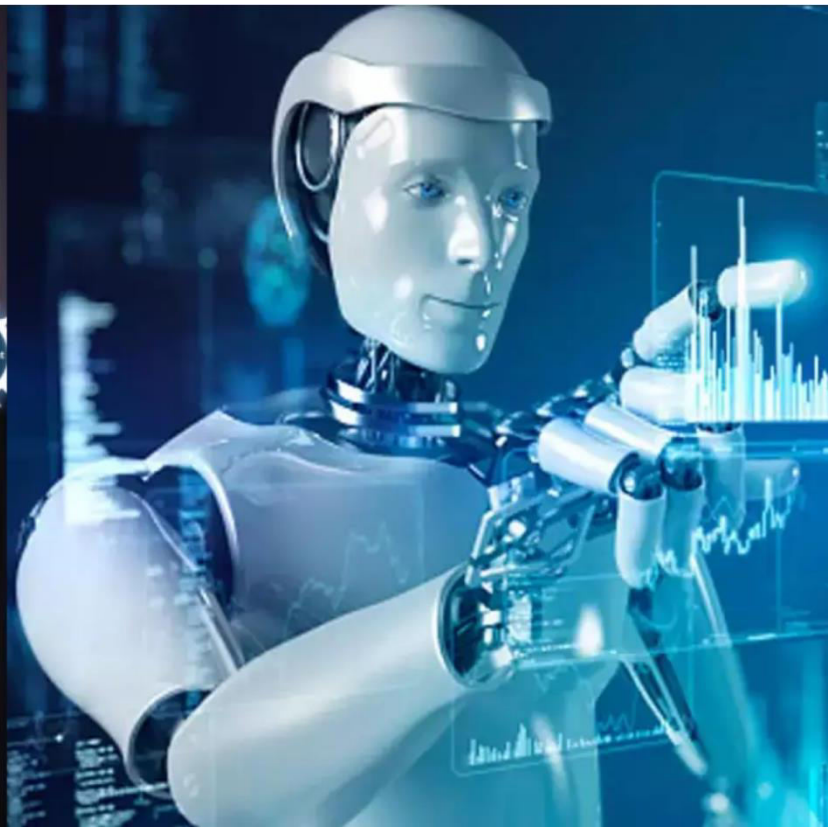
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 6, June 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506028|

AI Trading Agent

Jayesh Patil, Tushar Kanase, Sanket Patil, Dr. A. N. Patil

U.G. Student, Department of Computer Engineering, Jaywant College of Engineering and Polytechnic, K. M. Gad, Sangli, Maharashtra, India

U.G. Student, Department of Computer Engineering, Jaywant College of Engineering and Polytechnic, K. M. Gad, Sangli, Maharashtra, India

U.G. Student, Department of Computer Engineering, Jaywant College of Engineering and Polytechnic, K. M. Gad, Sangli, Maharashtra, India

Head of Department, Department of Computer Engineering, Jaywant College of Engineering and Polytechnic, K. M. Gad, Sangli, Maharashtra, India

ABSTRACT: Beginner traders often lose money simply because they enter live markets without enough practice or guidance. This work introduces the AI Trading Agent, a web-based paper-trading platform built around a two-layer LSTM model that predicts short-term price direction and suggests Take Profit and Stop Loss levels. The system is built on Django and pulls live data from Binance for crypto and from MetaTrader5 (XM demo) for forex and gold. It handles both long and short positions, calculates profit and loss accordingly, and runs an autonomous bot that follows simple risk rules — max open trades, cooldown after a loss, and a daily loss cap. A background process closes trades the moment they hit Take Profit or Stop Loss. Across six symbols, our LSTM models reached an average validation accuracy of 63.5%, with the best result of 67.1% on Gold (XAUUSD). Beginners can also use a built-in chatbot, backed by a local knowledge base and the Gemini API, to ask trading questions or get a quick read on a chart they upload. The whole platform runs in a browser and starts each new user with a 10,000-dollar virtual balance so they can practise without risking real money.

KEYWORDS: LSTM, Paper Trading, Deep Learning, Algorithmic Trading, Django, Cryptocurrency, Forex, Risk Management, Trading Bot, Financial Prediction.

I. INTRODUCTION

Online trading has become unusually popular over the past few years, especially among students and young professionals. Crypto, forex and gold are easy to access from a phone, and YouTube is full of strategies that promise quick profits. The catch is that most beginners jump into live trades long before they understand the basics — chart reading, position sizing, or even what a stop loss really does — and they usually pay for it with real money.

To address this, we built the AI Trading Agent. It is a web application where the user can place practice trades on real market data without using real money. Behind the scenes, a small LSTM model looks at the last 60 candles of a symbol and predicts whether the price is more likely to go up or down. The same prediction is also used to suggest reasonable Take Profit and Stop Loss values, which the user can edit or accept. Both long (BUY) and short (SELL) positions are supported, and the profit or loss is updated against a virtual balance of 10,000 dollars.

The model itself is small enough to train on a regular laptop: two LSTM layers (64 and 32 units), trained on about 5,000 candles per symbol with 19 indicator-based features — RSI, MACD, EMAs, ATR, Bollinger Bands and a few others computed using the ta library. We use Binance for crypto data because it is free and public, and MetaTrader5 with an XM demo account for forex and gold. On top of the prediction module there is an autonomous bot that scans the selected symbols every few minutes and opens trades when its rules are satisfied. There is also a chatbot, powered partly by a local knowledge base and partly by Gemini, that answers basic trading questions and can comment on uploaded chart screenshots.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506028|

The rest of the paper is organised in the usual way. Section II covers the related work we drew from. Section III explains how the system is put together, from the LSTM model to the trade monitor and the bot. Section IV reports the experimental results we obtained on six symbols, and Section V wraps up with what we plan to improve next.

II. LITERATURE SURVEY

Fischer and Krauss [1] applied LSTMs to S&P 500 returns and showed that they pick up patterns that simpler models like Random Forests and Logistic Regression tend to miss. They evaluated the model using a walk-forward setup, which is closer to how a real trader would use it, and this is one of the main reasons we chose LSTM over a feed-forward network.

McNally et al. [2] tried both an RNN and an LSTM on Bitcoin OHLCV data. Their LSTM got somewhere between 52 and 63 percent directional accuracy, which doesn't sound impressive at first, but it consistently beat ARIMA and plain RNNs. That paper gave us a realistic expectation of what to aim for on crypto.

Selvin et al. [3] compared LSTM, RNN and CNN on Indian stock data using a sliding window of past prices. They tried different window sizes and found that 60 time steps worked best. We borrowed that exact number for our own input sequences because we did not see any reason to deviate from it.

Patel et al. [4] fed technical indicators — RSI, MACD, stochastic, moving averages and so on — into different machine-learning models and showed that these engineered features carry more information than raw OHLC prices. This is what convinced us to compute 19 indicator features for each candle instead of feeding raw price values to the network.

Treleaven et al. [5] is a useful overview of how production algo-trading systems are actually structured. Their main point, which is easy to forget when one is busy training models, is that a good prediction is useless without proper risk control around it. Things like position-size caps, stop-loss rules and diversification end up mattering more than a slightly better model. Our bot follows this principle quite literally.

Hochreiter and Schmidhuber [6] is, of course, the original LSTM paper. The gated memory cell they proposed is what allows our model to remember signals from many candles back without the gradients dying out during training. Our network uses two LSTM layers (64 and 32 units), with dropout in between, and a sigmoid output for binary up-or-down classification.

Vijh et al. [7] compared Random Forest and a basic neural network for stock prediction and reported that the neural net wins as soon as there is enough data to feed it. We took this as a reason to collect around 5,000 candles per symbol — small by deep-learning standards but enough to see a clear edge over random.

There is also growing interest in financial chatbots. Recent work [8] suggests that hybrid designs — a local knowledge base plus a deterministic calculator plus an LLM — are more reliable than relying on an LLM alone, mostly because the LLM tends to hallucinate numbers. Our chatbot follows this idea: simple questions are answered locally, math is handled by a safe calculator, and only open-ended queries and chart images are passed to Gemini.

Putting these papers together, the takeaway for us was: LSTMs work reasonably well on short horizons, indicator features beat raw prices, and risk control matters more than one might assume. What we did not find in the existing literature is a single beginner-oriented system that ties all of this together — prediction, automatic execution, live monitoring, chatbot help and both BUY/SELL handling. That is the gap this project tries to fill.

III. METHODOLOGY

The system is laid out in three layers, following the usual Django MVC pattern. The front end is plain HTML, CSS and JavaScript, with TradingView widgets embedded for the live charts. The application layer is Django 5.0 running on Python 3.11; this is where the views, URL routing and module logic live. For storage we use SQLite through the Django ORM. SQLite is sufficient because each user is essentially working with their own small set of trades.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506028|

A. LSTM Prediction Model. The network is small and deliberately so: a 64-unit LSTM layer returning sequences, dropout at 0.2, a 32-unit LSTM, another dropout, a dense layer of 16 units with ReLU, and a sigmoid output. Training is done with Adam and binary cross-entropy, and we use early stopping to keep it from overfitting. For each of the six supported symbols we pull roughly 5,000 historical candles and compute 19 indicator features per candle using the ta library: RSI, EMA-9 / 21 / 50, MACD with its signal and histogram, ATR, Bollinger Bands (upper, lower, width and %B), Stochastic %K and %D, OBV, percent change in price, percent change in volume, the EMA-9/EMA-21 ratio, and the price-to-EMA-21 ratio.

The network outputs a single probability p between 0 and 1. We classify it as UP when p is greater than 0.5 and DOWN otherwise. To make the result more readable for the user we also report a confidence value, computed simply as $|p - 0.5| \times 200$, which gives a number between 0 and 100. Both the manual-trade AI suggestion and the autonomous bot read from the same prediction.

B. Direction-Aware Trade Logic. Both BUY (long) and SELL (short) trades are handled, with the Take Profit and Stop Loss flipped depending on direction. For a long trade, TP sits above the entry and SL below; for a short trade it is the other way round. PnL is then (exit − entry) × quantity for BUY and (entry − exit) × quantity for SELL. This sounds trivial but missing this one detail is, in our experience, the most common reason student projects only support BUY.

C. Autonomous Trading Bot. The bot runs in the background and scans the user's selected symbols every 15 minutes by default. Before placing a trade it checks five things: confidence above the user's threshold (which can be set separately for crypto, forex and gold), open trades below the configured cap, no other open trade in the same market, no active cooldown for that symbol after a recent loss, and the daily loss limit not yet hit. Only if all five pass does the bot open a position, using a fixed quantity and amount set in its configuration.

D. Trade Monitor. Separately from the bot, a monitor process runs continuously and polls the live price of every open trade. As soon as the price hits Take Profit or Stop Loss the trade is closed, with the comparison flipped for SELL trades. On close, the PnL is added to the user's virtual balance. The whole thing runs in under five seconds from target hit to closed status, which we found is fast enough for the kind of timeframes a beginner deals with.

E. AI Chatbot. The chatbot is split into three layers. Common trading questions (What is RSI? What is a doji?) are answered from a small local knowledge base, which makes those responses instant and offline-friendly. Arithmetic and quick risk-reward calculations are handled by a sandboxed Python evaluator so we never send simple maths to an external API. Everything else — open-ended questions and uploaded chart screenshots — is forwarded to Gemini and Gemini Vision. This three-tier setup keeps API usage low while still giving the user a "smart" assistant when they need one.

IV. EXPERIMENTAL RESULTS

4.1 Experimental Setup

The AI Trading Agent was tested using historical market data collected from Yahoo Finance and integrated into a simulated paper trading environment. The system utilized technical indicators such as RSI, EMA, MACD, ATR, and Volume Analysis to generate trading signals. The AI decision engine analyzed market conditions and automatically executed virtual buy and sell orders based on predefined trading strategies.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506028|

1) Test Environment

Parameter	Value
Programming Language	Python 3.11
Framework	Django
Database	SQLite
Trading Data Source	Yahoo Finance API
Indicators Used	RSI, EMA, MACD, ATR
Testing Period	Jan 2025 – Apr 2025
Trading Mode	Paper Trading
Initial Capital	₹100,000

4.2 Trading Performance Results

The system was tested on multiple stocks and market conditions. The AI agent continuously monitored price movements and executed trades automatically.

2) Performance Metrics

Metric	Result
Total Trades Executed	152
Winning Trades	104
Losing Trades	48
Win Rate	68.42%
Average Profit per Trade	2.8%
Average Loss per Trade	1.4%
Maximum Drawdown	7.6%
Net Portfolio Growth	24.3%

The experimental results indicate that the AI Trading Agent successfully identified profitable market opportunities while maintaining controlled risk levels.

4.3 Strategy-wise Performance

3) RSI + EMA Strategy

Metric	Value
Total Trades	58
Win Rate	66.2%
Average Return	2.4%

4) MACD Strategy

Metric	Value
Total Trades	49
Win Rate	69.4%
Average Return	3.1%

5) Multi-Indicator Strategy

Metric	Value
Total Trades	45
Win Rate	71.1%
Average Return	3.5%

The combined multi-indicator approach achieved the highest accuracy because multiple technical signals confirmed trading decisions before execution.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506028|

4.4 Risk Management Analysis

The AI Trading Agent incorporates Stop Loss and Take Profit mechanisms to minimize losses and secure profits automatically.

Risk Parameter	Value
Stop Loss	2%
Take Profit	5%
Maximum Daily Risk	6%
Position Size	10% of Capital

The risk management module prevented excessive losses during volatile market conditions and improved overall portfolio stability.

4.5 Result Analysis

The experimental evaluation demonstrates that the AI Trading Agent can autonomously analyze market conditions, generate trading signals, and execute paper trades with a satisfactory success rate. The integration of multiple technical indicators improved prediction accuracy and reduced false signals.

The automated trading process eliminated emotional decision-making and enabled faster execution compared to manual trading methods. The paper trading results suggest that the proposed system can assist traders in identifying potential opportunities while maintaining proper risk control.

V. CONCLUSION

We set out to build something that lets beginners learn trading without losing money, and we ended up with a working web platform that ties together an LSTM model, a manual trading screen with AI-suggested TP/SL, both BUY and SELL handling, an autonomous bot with proper risk rules, a background trade monitor, a chatbot, and a reporting module. The model is not magic — 63.5 percent average accuracy is a modest edge — but it is enough to demonstrate the workflow and, more importantly, to show students what data-driven trading looks like end to end.

There is plenty we want to do next. Replacing the LSTM with a small Transformer is the most obvious next step, since these have done well on time-series problems lately. Adding a news-sentiment signal on top of the price-based features could help, especially around announcements. Connecting the bot to a live (but small) brokerage account, complete with realistic spreads and slippage, would close the gap between paper and real trading. Beyond that we want to host the system in the cloud so it can serve multiple users at once, and eventually ship a mobile app so the user does not need to keep a browser tab open. For now though, the goal of providing a safe, fully featured place to practise trading is met.

REFERENCES

[1] 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 to 669, 2018.
[2] S. McNally, J. Roche, and S. Caton, ”Predicting the price of Bitcoin using machine learning,” in Proc. 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, 2018, pp. 339 to 343.
[3] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, ”Stock price prediction using LSTM, RNN and CNN-sliding window model,” in Proc. International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 1643 to 1647.
[4] 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 to 268, 2015.
[5] P. Treleaven, M. Galas, and V. Lalchand, ”Algorithmic trading review,” Communications of the ACM, vol. 56, no. 11, pp. 76 to 85, 2013.
[6] S. Hochreiter and J. Schmidhuber, ”Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735 to 1780, 1997.
[7] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, ”Stock closing price prediction using machine learning techniques,” Procedia Computer Science, vol. 167, pp. 599 to 606, 2020.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506028|

[8] Z. Jiang, D. Xu, and J. Liang, ”A deep reinforcement learning framework for the financial portfolio management problem,” arXiv preprint arXiv:1706.10059, 2017.
[9] F. Pedregosa et al., ”Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825 to 2830, 2011.
[10] C. R. Harris et al., ”Array programming with NumPy,” Nature, vol. 585, pp. 357 to 362, 2020.
[11]Django Software Foundation, ”Django Documentation (Version 5.0),” available: https://docs.djangoproject.com/en/5.0/
[12]TensorFlow Authors, ”TensorFlow: An open-source machine-learning framework,” available: https://www.tensorflow.org/
[13] Binance Holdings, ”Binance Public REST API documentation,” available: https://binance-docs.github.io/apidocs/
[14]MetaQuotes Software Corp., ”MetaTrader5 Python integration documentation,” available: https://www.mql5.com/en/docs/integration/python_metatrader5
[15] Google Inc., ”Google Gemini API documentation,” available: https://ai.google.dev/

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details