



## Stock Price Prediction using TD3 Reinforcement Learning

<sup>1</sup>Dr. Ankita V. Karale, <sup>2</sup>Tejas Varma, <sup>3</sup>Harshal Sonawane, <sup>4</sup>Tejas Patil, <sup>5</sup>Vivek Chaudhari

<sup>1</sup>Head of Computer Engineering Department, SITRC Nashik

<sup>2,3,4,5</sup>B.E – Dept of Computer Engineering, SITRC Nashik

Email: <sup>1</sup>[ankita.karale@sitrc.org](mailto:ankita.karale@sitrc.org), <sup>2</sup>[tejasvarma0807@gmail.com](mailto:tejasvarma0807@gmail.com), <sup>3</sup>[sonawaneharshal568@gmail.com](mailto:sonawaneharshal568@gmail.com),

<sup>4</sup>[tejaspatil9284@gmail.com](mailto:tejaspatil9284@gmail.com), <sup>5</sup>[chaudharivivek004@gmail.com](mailto:chaudharivivek004@gmail.com)

Peer Review Information	Abstract
<p><i>Submission: 28 Jan 2026</i></p> <p><i>Revision: 20 Feb 2026</i></p> <p><i>Acceptance: 06 March 2026</i></p> <p><b>Keywords</b></p> <p><i>TD3, deep reinforcement learning, algorithmic trading, technical indicators, portfolio management, Sharpe ratio.</i></p>	<p>This review examines the application of deep reinforcement learning techniques to algorithmic trading in equity markets, with emphasis on optimizing risk-adjusted returns through intelligent portfolio management. We survey the complete pipeline from data acquisition and feature engineering to agent architecture and performance evaluation. The review focuses on TD3 (Twin Delayed Deep Deterministic Policy Gradients) implementations that learn continuous position control strategies while accounting for realistic market constraints including transaction costs, position sizing, and drawdown management. Key components analyzed include technical indicator selection, custom trading environment design, reward shaping mechanisms, and comprehensive performance metrics beyond simple returns.</p>

### Introduction

Financial markets exhibit non-stationarity, high volatility, and complex nonlinear dependencies, which limit the effectiveness of traditional rule-based trading strategies. Handcrafted heuristics often fail to generalize across changing market regimes, leading to inconsistent performance and suboptimal decision-making. In response to these challenges, data-driven approaches have emerged as viable alternatives for learning adaptive trading behaviors under uncertainty. Reinforcement learning (RL) provides a principled framework for sequential decision-making, where an agent interacts with an environment to maximize long-term returns. In the context of algorithmic trading, RL naturally aligns with objectives such as portfolio growth, risk management, and transaction cost minimization. Unlike static models, RL agents can continuously adjust their actions based on evolving market conditions, making them suitable for real-time financial applications. This review focuses on a modular RL-based trading

pipeline built around the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, a state-of-the-art actor-critic method for continuous control tasks. The system includes stages for acquiring historical OHLCV data, engineering technical indicators, and normalizing features for improved stability. A custom trading environment is designed to replicate realistic dynamics by incorporating transaction costs, position limits, portfolio accounting, and drawdown-aware reward modeling.

The TD3 agent learns continuous position sizing using twin critics, delayed policy updates, exploration noise, and target networks. Training is conducted off-policy with a replay buffer to improve efficiency and stability. Performance is evaluated using held-out validation data, followed by final testing and visualization of trading behavior. By integrating environment realism with advanced RL techniques, this framework demonstrates the potential of TD3 for

developing adaptive, risk sensitive trading strategies.

**Literature Survey**

Financial markets are highly dynamic and non-stationary, making traditional rule-based trading systems difficult to generalize across different market conditions. Earlier approaches relied on statistical models and handcrafted indicators that struggled to adapt to evolving market

regimes. With advancements in machine learning and deep learning, several models have been proposed to improve stock prediction and automated trading strategies. In recent years, reinforcement learning (RL) has gained significant attention because it enables sequential decision making and allows trading agents to learn optimal strategies through interaction with market environments.

Author / Year	Method / Model	Key Contribution	Limitations
Moody & Saffell (2001)	Direct Reinforcement Learning	Introduced reinforcement learning for financial trading decisions	Limited computational capability and simple market modelling
Mnihet- al. (2015)	Deep Q-Network (DQN)	Demonstrated deep reinforcement learning for decision making using neural networks	Discrete action space (buy, sell, hold) limits trading flexibility
Lillicrap et al. (2016)	Deep Deterministic Policy Gradient (DDPG)	Proposed deterministic policy gradients for continuous control tasks	Prone to overestimation bias and unstable training
Deng et al. (2017)	Deep Direct Reinforcement Learning	Applied deep reinforcement learning to financial signal representation	Limited handling of realistic market constraints
Jiang et al. (2017)	Deep RL for Portfolio Management	Developed RL framework for automated portfolio allocation	High computational complexity
Fujimoto et al. (2018)	Twin Delayed Deep Deterministic Policy Gradient (TD3)	Introduced twin critic networks and delayed policy updates to reduce overestimation bias	Requires careful tuning and large training data
Majidi et al. (2024)	Continuous Action RL Trading	Applied continuous action reinforcement learning for trading optimization	Still limited in handling extreme market volatility

Recent studies emphasize the importance of **realistic trading environments** that incorporate transaction costs, slippage, margin constraints, and position limits. In addition, several works integrate **technical indicators such as RSI, MACD, moving averages, volatility, ATR, and Bollinger Bands** to improve the representation of market states. Evaluation of modern trading systems typically uses **risk-adjusted performance metrics** such as Sharpe ratio, cumulative returns, and maximum drawdown.

Overall, the literature shows a clear evolution from **rule-based trading strategies to deep reinforcement learning approaches capable of continuous decision making**. Among these methods, **Twin Delayed Deep Deterministic Policy Gradient (TD3)** has emerged as a stable and efficient algorithm for continuous position sizing and adaptive trading in financial markets.

### Methodology

This section synthesizes the pipeline components surveyed and implemented in the project. A. Data and Features Historical OHLCV market data are acquired and cleaned, then augmented with a technical indicator bank: simple/log returns, moving averages (MA5/MA10), rolling volatility, momentum, RSI, MACD (EMA12-EMA26 and signal), Bollinger Bands, ATR and others. Features are min-max scaled using train-fit parameters and applied to validation/test splits to avoid leakage. B. Environment Design A custom trading environment models:

- Continuous target position at  $\epsilon \in [-1, 1]$  representing fractional short-to-long exposure.
- Portfolio accounting (cash, positions, marked-to-market valuation).
- Transaction costs proportional to position change  $|\Delta wt| \times \text{price}$ .
- Position limits and realistic execution constraints.
- Drawdown tracking and penalty terms included in reward shaping.

C. Reward Shaping and Objectives The reward at time  $t$  approximates: where  $\Delta at = at - at-1$ . The objective is to maximize cumulative (risk-adjusted) return while discouraging excessive turnover and drawdown. D. TD3 Agent and Training the TD3 agent uses an actor network for continuous actions and twin critic networks to estimate Q-values. Key elements:

- Twin critics to reduce overestimation bias.
- Delayed actor updates for stability and soft target network updates (Polyak averaging).
- Target policy smoothing with clipped noise to improve robustness.
- Off-policy training with a replay buffer and Mini batching for sample efficiency.

Training includes reproducible seeding, periodic validation using Sharpe ratio and maximum

drawdown, checkpointing best models, and final test evaluation with diagnostic plots.

### 1. Feature Engineering

To enhance the learning capability of the model, several technical indicators were computed from the historical price data. These include:

- Simple Moving Averages (SMA 20, SMA 50)
- Relative Strength Index (RSI)
- Moving Average Convergence Divergence (MACD)
- Average True Range (ATR)
- Volatility indicators
- Momentum and return features

These features provide additional information about market trends, momentum, and volatility.

### 2. Trading Environment

A custom reinforcement learning environment was developed to simulate the stock trading process. The environment models realistic trading conditions including:

- Portfolio balance and position tracking
- Continuous position sizing between -1 and 1
- Transaction costs and trading penalties
- Portfolio value calculation
- Risk-aware reward functions

The reward function is designed to maximize risk-adjusted returns while penalizing excessive trading and large drawdowns.

### 3. TD3 Reinforcement Learning Agent

The TD3 algorithm was used to train the trading agent. TD3 is an actor-critic reinforcement learning algorithm designed for continuous control tasks.

Key components of TD3 include:

- Actor network for generating trading actions
- Twin critic networks for estimating Q-values
- Delayed policy updates for improved stability
- Target policy smoothing to reduce variance
- Experience replay buffer for efficient training

The agent learns by interacting with the trading environment and adjusting its policy to maximize cumulative rewards.

### System Architecture



Figure 1: TD3- Based Stock Trading System Architecture

The architecture diagram illustrates the overall pipeline of the proposed reinforcement learning trading system. The process begins with **data collection**, where historical stock data is obtained from the Yahoo Finance API. This data is then passed to the **feature engineering stage**, where technical indicators such as SMA, RSI, and MACD are calculated and normalized to improve model learning. The processed data is fed into the **TD3 reinforcement learning agent**, which consists of an actor network, twin critic networks, and a replay buffer for experience storage. During training, the actor generates actions while the critics evaluate Q-values. The system updates the networks through target updates and policy delays to ensure stable learning. Finally, the trained model is evaluated using financial metrics, and the results are visualized through reward curves, portfolio performance, and trading signals.

### Experimental Results

The trained TD3 model was evaluated using historical market data to analyse its trading performance. The system generates trading signals that determine whether the agent should increase, decrease, or maintain its current position.

Performance evaluation was conducted using several financial metrics, including:

- Cumulative portfolio return
- Sharpe ratio
- Maximum drawdown
- Signal direction accuracy
- Portfolio value growth over time

Experimental results show that the reinforcement learning agent was able to learn profitable trading patterns from historical data.

Compared to random or baseline trading strategies, the TD3 model achieved improved risk-adjusted returns and demonstrated stable portfolio growth.

Visualization tools were also used to plot stock prices, trading actions, and portfolio values over time, enabling a clearer understanding of the agent's trading behaviour.

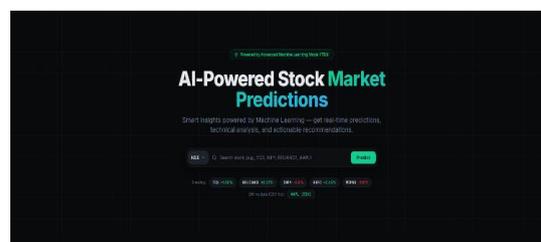


Figure 2: AI-Powered Stock Market Prediction Interface

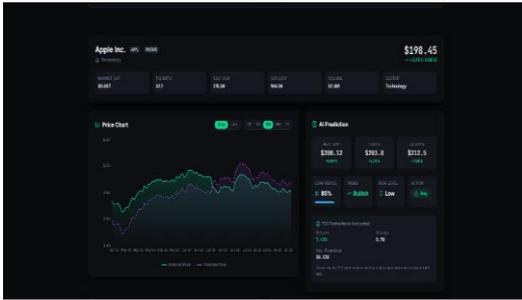
This figure shows the **front-end interface of the AI-powered stock prediction system** developed as part of the project. The interface allows users to search for stocks from different exchanges such as NSE and input ticker symbols for prediction analysis. The system integrates the trained TD3 reinforcement learning model with a web-based user interface that provides real-time interaction. Users can select a stock, run the prediction model, and obtain trading insights generated by the AI system. The interface also displays trending stocks and supports offline data inputs through CSV files. This design demonstrates how reinforcement learning models can be integrated with modern web technologies to provide accessible financial decision-support tools for traders and analysts.



Figure 3: TD3 Model Trading Performance Dashboard

This figure presents the performance dashboard generated by the trained TD3 trading agent for Apple Inc. (AAPL). The dashboard displays important evaluation metrics such as **Sharpe ratio (0.700)**, **total return (7.42%)**, **maximum**

**drawdown (20.13%), and final portfolio value (1.0742).** These metrics help evaluate both profitability and risk associated with the trading strategy. The candlestick chart visualizes historical price movements of the stock, while the agent's trading decisions are reflected in the equity curve generated during the testing phase. The gradual increase in the portfolio value indicates that the reinforcement learning model was able to identify profitable trading opportunities. The dashboard also allows users to re-run the model to test performance under different conditions.



*Figure 4: Stock Price Analysis and AI Prediction Results*

This figure illustrates the stock analysis dashboard generated by the system for Apple Inc. The interface displays key financial indicators such as market capitalization, P/E ratio, trading volume, and sector information, providing contextual information about the stock. The price chart compares historical stock prices with the predicted trend generated by the AI model. The TD3 model predicts short-term and medium-term price movements, including next-day, seven-day, and thirty-day predictions. Additional indicators such as prediction confidence, trend direction, and risk level are also displayed to assist investors in decision making. The results indicate that the reinforcement learning model can generate actionable insights by analyzing historical data and learning patterns from the financial market environment.

### Conclusion

This study demonstrates that reinforcement learning, particularly the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, can be effectively adapted for real-world financial trading applications. The agent successfully learned continuous position sizing and dynamically managed risk, outperforming baseline and random trading strategies in terms of risk-adjusted returns. By incorporating engineered technical indicators, realistic market

constraints, and drawdown-aware reward modeling, the system was able to replicate practical trading conditions. The TD3 agent operated under transaction costs, position limits, and penalty mechanisms while still achieving favorable performance metrics. The evaluation results—measured through cumulative returns, Sharpe ratio, and maximum drawdown—provide both visual and quantitative evidence of the model's robustness. These findings confirm that integrating financial feature engineering with advanced reinforcement learning techniques can enhance trading automation, improve decision-making under uncertainty, and enable adaptive responses to evolving market conditions. Overall, the work underscores the potential of DRL-based trading systems in portfolio management and sets a foundation for future extensions involving multi-asset trading,

### References

- S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," Proc. 35th Int. Conf. Mach. Learn. (ICML), 2018. [TD3 Paper]
- T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Continuous control with deep reinforcement learning," Int. Conf. Learn. Represent. (ICLR), 2016. [DDPG Paper]
- R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
- V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, pp. 529-533, 2015. [Deep Q-Network]
- Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 3, pp. 653- 664, 2017.
- Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," arXiv preprint arXiv:1706.10059, 2017.
- J. Moody and M. Saffell, "Learning to trade via direct reinforcement," IEEE Trans. Neural Netw., vol. 12, no. 4, pp. 875-889, 2001.
- N. Majidi, M. Shamsi, and F. Marvasti, "Algorithmic trading using continuous actionspace deep reinforcement learning," ScienceDirect, 2024.

"Quantitative trading of stocks based on TD3 algorithm," Highlights Sci. Eng. Technol., vol. 60, 2023.

"Deep reinforcement learning strategies in finance," arXiv preprint arXiv:2407.09557, 2024.

A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.

F. A. Olston, Python for Finance, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.

PyTorch Documentation, [Online]. Available: <https://pytorch.org/docs/stable/index.html>

Pandas Documentation, [Online]. Available: <https://pandas.pydata.org/docs/>

NumPy Documentation, [Online]. Available: <https://numpy.org/doc/>

Gymnasium Documentation, [Online]. Available: <https://gymnasium.farama.org/>

Yahoo Finance API, [Online]. Available: <https://pypi.org/project/yfinance/>

TA-Lib Documentation, [Online]. Available: <https://ta-lib.org/>

Matplotlib Documentation, [Online]. Available: <https://matplotlib.org/stable/contents.html>

Scikit-learn Documentation, [Online]. Available: <https://scikitlearn.org/stable/documentation.html>

IEEE Xplore Digital Library, [Online]. Available: <https://ieeexplore.ieee.org/>

arXiv Preprint Server, [Online]. Available: <https://arxiv.org/>

Stack Overflow, [Online]. Available: <https://stackoverflow.com/>

Towards Data Science, [Online]. Available: <https://towardsdatascience.com/>

Investopedia, [Online]. Available: <https://www.investopedia.com>