# Enhancing Stock Trading Performance with Deep Q-Learning by Addressing Noisy Data through Advanced Denoising Techniques

Soroush Shahsafi
*Concordia Institute for Information System Engineering*
*Concordia University*
Montreal, Canada
soroush.shahsafi@concordia.ca

Farnoosh Naderkhani
*Concordia Institute for Information System Engineering*
*Concordia University*
Montreal, Canada
farnoosh.naderkhani@concordia.ca

***Abstract*—This study presents a comparative analysis of different reinforcement learning configurations for the stock trading problem, with IBM as a case study. To address the challenge of noisy data, we explore the effectiveness of various denoising methods, including Wavelet Transform, Temporal Attention Network (TAN), and Fourier Transform, in improving model performance. We employ two different architectures, Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM), to calculate Q-values for each possible action, resulting in six distinct configurations. Evaluation is based on key metrics such as yearly returns, Sharpe ratios, and maximum drawdowns over a specified timeframe. We compare the performance of our models against benchmark strategies including Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and Bollinger Bands. Our results demonstrate that DQN-based trading outperforms benchmark methods. Furthermore, configurations utilizing TAN, whether in conjunction with MLP or LSTM, consistently exhibit superior performance. These findings suggest that TAN-based denoising methods combined with DQN offer promising solutions for enhancing stock trading strategies using reinforcement learning techniques.**

***Index Terms*—Reinforcement Learning, DQN, Stock Trading, Denoising Methods, Temporal Attention Network**

## I. Introduction

In recent years, machine learning techniques, especially reinforcement learning (RL), have attracted considerable attention in algorithmic trading. Traditional supervised learning approaches have been extensively utilized in forecasting stock prices based on historical data. However, merely predicting price movements does not guarantee profitable trading decisions. Also, they often struggle to grasp the dynamic nature of financial markets, which might not result in optimal trading decisions. Successful trading requires dynamic adaptation to market conditions. This motivation has led to the exploration of RL techniques for learning optimal trading policies.

RL has become a game-changing technique in the finance and stock trading domains, providing novel solutions to the complex problems of this fast-paced field. This machine learning paradigm has been applied to financial markets to optimize trading methods, portfolio management, risk mitigation, and decision-making procedures. Its flexibility, capacity to learn from experiences and ability to optimize cumulative rewards make it significant [1]. Success in the financial industry depends on the ability to make well-informed decisions efficiently. Conventional trading methods often depend on pre-established rules, which might not take into consideration the complex, changing nature of financial markets. This is where RL enters the picture as an invaluable instrument that helps investors and traders better manage the complex dynamics of the stock markets [2]. Reinforcement learning provides a promising way to develop successful trading behaviors by refining trading strategies to get the most rewards over time. RL aims to maximize long-term rewards by continuously interacting with the environment and adapting its approach accordingly [3].

Deep reinforcement learning (DRL) further extends RL by incorporating deep neural networks to handle high-dimensional input spaces and complex decision-making tasks. By leveraging the representational power of neural networks, DRL agents can learn intricate patterns from raw input data and make informed trading decisions. However, implementing DRL for algorithmic trading presents several challenges. Financial time series data are often noisy and highly non-stationary due to various factors such as market manipulation, news events, and investor behaviors. So, designing effective trading strategies involves addressing issues such as adapting to changing market conditions [4].

To tackle these challenges, researchers have proposed various approaches to preprocess financial data and devise robust trading strategies using DRL. One such approach involves employing denoising methods to remove noise from financial time series data before inputting it into DRL models [5].

The contribution of our paper lies in its exploration of DRL applied to a single stock trading problem. We begin this exploration by highlighting the crucial step of preprocessing input data to remove noise, which is essential for improving the robustness of learning algorithms. Particularly, we examine various denoising techniques, such as wavelet transformation (WT), Temporal Attention Network (TAN), and Fourier transform, aiming to assess their impact on trading performance within the DRL framework. Furthermore, our study extends to comparing the effectiveness of different networks, including Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM) as agent architectures. Through these comparative analyses, we aim to identify the most effective combination of the agent network and denoising technique, thus offering valuable insights for future research and practical applications in financial markets.

The remainder of this paper is organized as follows: Section II reviews related work, highlighting the application of reinforcement learning in trading strategies and data-denoising methods. Section III describes the methodology employed in this study, including details of the DQN and denoising techniques. Section IV presents the experimental results and performance comparisons. Finally, Section V concludes the paper and outlines directions for future research.

## II. Background and Related Work

Studies have explored how DRL algorithms can enhance trading strategies in various market scenarios. Liang et al. [6] found Policy Gradient (PG) to outperform other algorithms in portfolio management. Zhang et al. [7] demonstrated the effectiveness of RL-based strategies in developing trading strategies for continuous futures contracts. The practical approach proposed by Xiong et al. [8] showcases the efficacy of Deep Deterministic Policy Gradient (DDPG) agents in stock trading, outperforming traditional portfolio allocation methods. Ensemble strategies like the one presented in [9] leverage multiple RL algorithms to develop robust trading frameworks, offering better performance over traditional methods. Furthermore, the framework by Li et al. [10] addresses challenges in feature extraction and trading strategy design, achieving stable returns in both stock and futures markets. They presented novel frameworks for practical algorithmic trading using deep robust reinforcement learning. By leveraging techniques like stacked denoising autoencoders (SDAEs) and long short-term memory (LSTM) units, these frameworks enable robust feature extraction and resolve dependencies in financial time series data. Experimental results demonstrate superior performance over baselines, highlighting the effectiveness of denoising techniques in improving trading strategy design and implementation. These findings highlight the potential of DRL in improving trading outcomes across different market environments.

In algorithmic trading, the quality of input data significantly impacts strategy effectiveness. Reducing noise and enhancing the quality of input data contributes significantly to the development of robust and reliable trading strategies. Shavandi and Khedmati [11] used an innovative way of dealing with noisy data. They introduced a multi-agent deep RL framework, which uses collective intelligence to outperform individual agents and traditional strategies across different trading timeframes, showcasing the versatility of DRL. Indeed, the expert agents at lower timeframes, in addition to benefiting from the details of price movements in their specific timeframe, can be more resistant to the noise of their timeframe as well.

Different denoising methods can be employed to address noisy data. Among these methods, Wavelet transformation has emerged as a powerful tool for denoising financial time series data. Several studies have showcased its effectiveness in enhancing trading system performance by reducing noise while preserving essential signal characteristics [12], [13]. Moreover, Padding-based Fourier Transform Denoising (P-FTD) has gained traction as an effective approach for mitigating noise in financial time series data [14].

This paper focuses on the advanced application of information fusion in stock trading, utilizing RL to transcend its conventional use of merely aggregating sensor data for accuracy. Instead, it aims to integrate market data, decision-making processes, and outcome-based learning into a cohesive framework. By leveraging key features like closing prices within a comprehensive information set, the approach fosters more informed and strategic trading decisions through the adept use of RL. While some studies have compared DRL strategies with traditional methods, there is still a lack of comprehensive analyses that systematically assess the impact of denoising techniques on trading performance using DRL architectures. This study seeks to address this gap by conducting a focused investigation into the efficacy of different denoising methods, including Wavelet Transform, Temporal Attention Network (TAN), and Fourier Transform, in enhancing the performance of DRL models for a single-stock trading problem.

## III. Material and Methods

### A. Reinforcement Learning Structure

The core of our approach lies in the Deep Q-Learning algorithm, which facilitates the learning of the action-value function $Q^*(s,a)$ by iteratively updating Q-values based on observed experiences. The agent interacts with the environment by selecting actions according to an $\varepsilon$-greedy policy, enabling exploration and exploitation of the action space. The Bellman equation governs the update process:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t)+\alpha\left(r_t+\gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)\right) \tag{1}$$

Here, $\alpha$ represents the learning rate, and $\gamma$ is the discount factor, balancing the influence of future rewards. The agent learns to optimize its decision-making policy by maximizing cumulative rewards over time.

*1) Action Space:* An action is the agent's response to the situation as it is at that moment. The agent's goal is to select actions that maximize its cumulative rewards over time. In this research, the action space comprises three actions: sell, hold, and buy, offering flexibility in decision-making. When the agent is going to trade and take action, there are some simplifying assumptions. Upon a sell decision, we promptly execute it by selling all shares of the specified stock, with the proceeds added to our available cash. For buying, if multiple stocks are selected for trading, we iteratively buy one share of each stock until we exhaust the available cash.

*2) State Space:* The state is an illustration of the status of the environment at a specific moment in time. A state in stock trading could contain data on past price movements, economic indicators, stock prices, and other relevant information. In this paper, the state space is represented as the denoised historical close prices, obtained after applying denoising methods. This state encapsulates crucial information about market dynamics, aiding the agent in making informed trading decisions.

*3) Reward Function:* The agent receives a numerical signal from the environment as a reward for doing a certain action in a particular condition. It measures the action's immediate cost or benefit. Rewards in stock trading could be correlated with the agent's trading decisions' gain or loss. Profitable acts are rewarded positively, and unprofitable actions are discouraged by negative rewards.

In this research, we define the reward function as the difference between the current portfolio value and its previous value. The portfolio value is computed using Equation 2, where $S_i$ represents the quantity of stock $i$ owned by the agent, $P_i$ represents the price of stock $i$, and $C$ represents the available cash in hand.

$$\text{Portfolio Value} = \sum_{i=1}^{n}(S_i \times P_i) + C \tag{2}$$

*4) The Agent Network:* To approximate the action-value function $Q(s, a)$, we employ two distinct DRL architectures: Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM). The inclusion of LSTM alongside MLP in our implementation serves to explore the impact of incorporating temporal dependencies in the data. LSTM, with its recurrent connections, can capture sequential patterns and long-term dependencies, potentially enhancing the agent's ability to discern market trends and make informed decisions. Through comparative analysis, we aim to elucidate the difference in model performance for each architecture.

*5) Replay Buffer:* The replay buffer serves as a crucial component in our model, storing experiences observed and facilitating sample efficiency in training. By randomly sampling experiences from the replay buffer, we mitigate issues related to correlated samples and stabilize the training process.

### *B. Denoising Methods*

One of the key challenges in time series analysis is dealing with noise, which can obscure underlying patterns and lead to inaccurate results.

*1) Wavelet Transformer:* Wavelet analysis, renowned for its effectiveness in fields like image and signal processing, has increasingly found application in economic and financial domains. Wavelet transform offers distinct advantages for resolving time series analysis challenges in financial contexts by enabling the decomposition and reconstruction of data across various time and frequency scales. The wavelet threshold denoising technique, a key aspect of Wavelet Transformer, operates by decomposing a signal into wavelet coefficients and selectively eliminating noise based on a chosen threshold. [15]

Practically, Wavelet Transformer operates by breaking down financial time series data into low-frequency and high-frequency components. By isolating and removing high-frequency noise components, the method aims to retain essential signal characteristics while preserving essential features of the original financial time series data, thus enhancing the robustness of deep reinforcement learning models in stock trading applications. This approach aligns with the necessity to filter out noise from financial data to improve model generalization and predictive accuracy. [16]

Furthermore, Wavelet Transformer benefits from the multi-scale analysis capabilities inherent in wavelet transforms. This enables the extraction of relevant features from financial time series data, providing valuable insights for informed decision-making in stock trading applications. [17]

Leveraging the Wavelet Transformer's ability to decompose data into different frequency scales, we first perform wavelet decomposition. This process involves breaking down the time series $x(t)$ into wavelet coefficients $c_{j,k}$ through the utilization of a chosen wavelet basis function $\psi_{j,k}(t)$, where $j$ denotes the scale and $k$ represents the translation. The decomposition process is represented mathematically as $c_{j,k} = \langle x(t), \psi_{j,k}(t)\rangle$. Following this, the threshold level $\tau$ is determined based on the statistical properties of the wavelet coefficients. Typically, the threshold is established as a fraction $\alpha$ of the maximum coefficient magnitude, denoted by $\tau = \alpha \cdot \max(|c_{j,k}|)$. Subsequently, soft thresholding is applied to the wavelet coefficients to eliminate noise while retaining significant signal components. Soft thresholding is mathematically represented as $c^{'}_{j,k} = \text{sign}(c_{j,k})\cdot\max(|c_{j,k}|-\tau, 0)$, where $c^{'}_{j,k}$ signifies the thresholded coefficient. This process selectively compresses the coefficients based on the threshold, removing high-frequency noise components while preserving essential signal features. Ultimately, the denoised signal $\hat{x}(t)$ is reconstructed from the thresholded wavelet coefficients using inverse wavelet transform, expressed as $\hat{x}(t) = \sum_{j,k} c^{'}_{j,k}\psi_{j,k}(t)$. This reconstructed signal serves as a denoised version of the original time series, suitable for subsequent analysis and modeling.

*2) Fourier Transform:* The Fourier Transform has been widely utilized across various domains for its ability to analyze signals in both the time and frequency domains. In the context of time series analysis, particularly in financial markets, Fourier Transform-based denoising techniques have gained attention due to their effectiveness in removing noise. The literature review reveals several studies, highlighting the importance of denoising financial time series data for stable model learning. Methods such as padding-based

Fourier Transform denoising (P-FTD) have been proposed to eliminate noise waveforms in the frequency domain, thereby enhancing the performance of predictive models [18]. Also, the application of Fourier Transform in signal denoising extends as a classical method, capable of extracting useful information from signals while removing noise. [19]

The Fourier Transform, denoted as $F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt$, plays a crucial role in isolating relevant signal components from noise. After applying the Fourier Transform to the noisy signal, a threshold is determined based on the characteristics of the signal and the desired level of noise removal. High-frequency components in the frequency domain above this threshold are then set to zero, effectively removing noise components from the signal spectrum. The Inverse Fourier Transform, represented as $f(t) = \int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega$, is subsequently applied to the modified frequency spectrum, converting the filtered frequency-domain signal back to the time domain and yielding the denoised signal. This process significantly enhances the robustness and reliability of subsequent analyses or predictive modeling tasks on financial time series data.

*3) Temporal Attention Network:* While the primary focus of Temporal Attention Networks is on forecasting future values, their attention mechanisms can implicitly aid in noise removal. By selectively attending to relevant temporal patterns and filtering out irrelevant information, TANs effectively denoise time series data. The proposed temporal pattern attention mechanism in [20] and the multimodal attention learning structure in [21] demonstrate the capability of TANs to remove noise while preserving essential signal components. The architecture consisted of an attention mechanism applied to the input sequence, allowing the network to focus on relevant temporal patterns while filtering out noise selectively. The Temporal Attention Network (TAN) employs attention mechanisms to selectively filter noise from time series data while preserving crucial temporal patterns. We trained the Temporal Attention Network using a set of input-output pairs generated from the normalized time series data. The network was trained using mean squared error loss and the Adam optimizer. We iteratively adjusted the network's parameters over multiple epochs to minimize the difference between the predicted and actual values.

In the context of TANs for denoising time series data, the input sequence undergoes normalization to zero mean and unit variance, ensuring uniform scale across the data. Following normalization, an attention mechanism is applied to the input sequence, producing attention weights that highlight relevant temporal patterns while suppressing noise. This attention mechanism is typically implemented as a weighted sum of the input features, where the weights are determined by the attention scores. By focusing on prominent temporal features, the network effectively denoises the data while preserving essential signal components. Training the TAN involves optimizing its parameters to minimize the mean squared error between the predicted and actual values, typically achieved through backpropagation and optimization algorithms like Adam. Once trained, the TAN can be utilized to denoise input sequences by passing them through the network and reconstructing the denoised output, thus offering a robust framework for handling noisy time series data in various applications such as stock trading.

## IV. Evaluating and Comparison of the Results

In this section, we present the evaluation results of our proposed deep Q-learning (DQN) models for the single-stock trading problem. The dataset used in this paper consists of historical close prices of IBM, one of the thirty companies listed on the Dow Jones Industrial Average (Dow30). The dataset provides a detailed record of IBM's daily close prices from November 2011 to December 2021, covering approximately ten years. As is shown in Figure 1 Data from November 2011 to December 2019 were used for model training, and data from January 2020 to December 2021 were used for testing. Table I demonstrates the statistical parameters of the data offering valuable insights into the behavior of IBM's stock prices during this specified period. The measures of skewness and kurtosis provide insights into the shape of the distribution. With skewness and kurtosis values of -0.26 and -0.27 respectively, the distribution appears to be slightly negatively skewed and exhibiting slight leptokurtic tendencies, indicating a relatively narrow peak compared to the normal distribution.

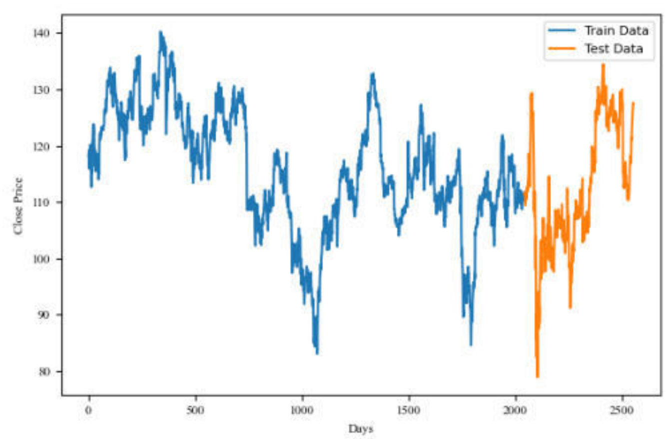


Fig. 1. IBM historical close prices: train and test dataset

We employed various denoising techniques, including WT, TAN, and Fourier Transform, to preprocess the data before feeding it into the models. Additionally, we compared the performance of two different agent networks: MLP and LSTM. Analyzing the yearly return, Sharpe ratio and maximum drawdown for each DQN configuration provides insights into risk-adjusted returns and downside risk. The Sharpe ratio is a mathematical indicator that measures the average excess return earned per unit of volatility above the risk-free rate. It offers insights into how greater volatility and risk in an investment strategy can correspond to higher excess returns over a specific period. The risk-free rate is assumed to be zero in this research.

TABLE I
SUMMARY OF STATISTICAL MEASURES

| Statistical Measure | Value |
|---|---|
| Mean | 115.13 |
| Standard Error | 0.21 |
| Median | 114.55 |
| Mode | 127.27 |
| Standard Deviation | 10.8 |
| Sample Variance | 116.61 |
| Kurtosis | -0.27 |
| Skewness | -0.26 |
| Range | 61.3 |
| Minimum | 78.91 |
| Maximum | 140.20 |

$$\text{Sharpe Ratio} = \frac{R_P - R_f}{\sigma_P} \quad (3)$$

Where:

$R_P$ : Portfolio return

$R_f$ : Risk-free rate

$\sigma_P$ : Standard deviation of the portfolio value

Also, Max drawdown is a metric that assesses the potential downside risk within a defined timeframe. It represents the largest observed loss from the highest point (peak) to the lowest point (trough) in the portfolio before making a full recovery and reaching a new peak.

$$\text{Max Drawdown} = \frac{\text{Trough value} - \text{Peak value}}{\text{Peak value}} \times 100 \quad (4)$$

Totally, six different configurations have been experimented with, each utilizing a combination of denoising methods (Wavelet Transformer, Temporal Attention Network, and Fourier) and two types of agent networks (Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM)). IBM's denoised historical close prices served as the state space for our reinforcement learning model. Table II presents the final investment value at the end of the test dataset, starting with an initial cash of $ 10,000.

TABLE II
FINAL INVESTMENT VALUES OF DIFFERENT DQN CONFIGURATIONS

| DQN Configurations | Investment Final Value |
|---|---|
| MLP-Fourier | $ 25202.53 |
| MLP-WT | $ 18667.7 |
| MLP-TAN | $ 26707.75 |
| LSTM-Fourier | $ 25181.36 |
| LSTM-WT | $ 18652.12 |
| LSTM-TAN | $ 26698.64 |

Also, the yearly results for the years 2020 and 2021 are presented in Table III and Table IV. In both years, the TAN-based configurations consistently outperformed the others in terms of all metrics, indicating the effectiveness of temporal attention mechanisms in capturing relevant temporal patterns for trading decisions. The Fourier-based configurations also demonstrated competitive performance.

TABLE III
PERFORMANCE METRICS FOR DIFFERENT CONFIGURATIONS IN 2020

| Configuration | Return | Sharpe Ratio | Maximum Drawdown |
|---|---|---|---|
| MLP-Fourier | 0.6534 | 39.992 | 0.1377 |
| MLP-WT | 0.3989 | 24.5687 | 0.1541 |
| MLP-TAN | 0.7259 | 41.1202 | 0.0771 |
| LSTM-Fourier | 0.6525 | 39.9145 | 0.1375 |
| LSTM-WT | 0.3971 | 24.4635 | 0.154 |
| LSTM-TAN | 0.7245 | 41.0628 | 0.0772 |

TABLE IV
PERFORMANCE METRICS FOR DIFFERENT CONFIGURATIONS IN 2021

| Configuration | Return | Sharpe Ratio | Maximum Drawdown |
|---|---|---|---|
| MLP-Fourier | 0.4674 | 54.3015 | 0.0503 |
| MLP-WT | 0.2818 | 24.1024 | 0.1034 |
| MLP-TAN | 0.5206 | 58.1485 | 0.0357 |
| LSTM-Fourier | 0.4671 | 54.2782 | 0.0503 |
| LSTM-WT | 0.2824 | 24.1541 | 0.1035 |
| LSTM-TAN | 0.5213 | 58.245 | 0.0357 |

To demonstrate the potential of DQN in stock trading, We evaluated the effectiveness of these models in comparison to the following traditional benchmark strategies, with an initial investment value of $10,000 in all scenarios:

- **Relative Strength Index (RSI)**: A momentum oscillator assessing the pace and magnitude of price shifts. It's commonly employed to detect overbought or oversold conditions within a market. In employing the RSI for trading, the approach involved utilizing the RSI values to generate buy and sell signals. When the RSI fall below the oversold threshold (set at 30), a buy signal will be initiated, indicating a potential opportunity for purchase. Conversely, when the RSI surpasse the overbought threshold (set at 70), a sell signal will be triggered, suggesting a possible time to sell. These signals are then used to simulate trading actions, with the algorithm buying upon a buy signal and selling upon a sell signal, with adjustments made accordingly to the capital and position. This method generated a result of $10,740.81 at the end of the test dataset.
- **Moving Average Convergence Divergence (MACD)**: A momentum indicator tracking trends, showcasing the correlation between two moving averages of a security's price. It serves to indicate shifts in the strength, direction, momentum, and duration of a trend in a stock's price. For the Moving Average Convergence Divergence (MACD) strategy, the focus is on the MACD line, signal line, and MACD histogram. Trading signals are generated based on the interactions between these components. Specifically, when the MACD histogram crosses above zero, a buy signal will be activated, indicating a potential bullish trend. Conversely, when the MACD histogram crosses below zero, a sell signal will be triggered, suggesting a potential bearish trend. This strategy yielded a result of

$13,191.37.

- **Bollinger Bands**: Bollinger Bands are utilized to identify potential overbought or oversold conditions and to assess market volatility. Regarding the Bollinger Bands approach, buy and sell signals are derived from the price movements of the upper and lower bands of the Bollinger Bands. A buy signal will be generated when the price crosses below the lower band, indicating a potential undervaluation and a buying opportunity. Conversely, a sell signal will be triggered when the price crosses above the upper band, indicating a potential overvaluation and a selling opportunity. Figure 2 demonstrates the buy and sell signals within the historical close prices of IBM. These signals are then employed to simulate trading actions, with the algorithm executing buys and sells accordingly based on the signals generated. Employing this strategy resulted in a value of $18,208.63 greater than the previous benchmark strategies.

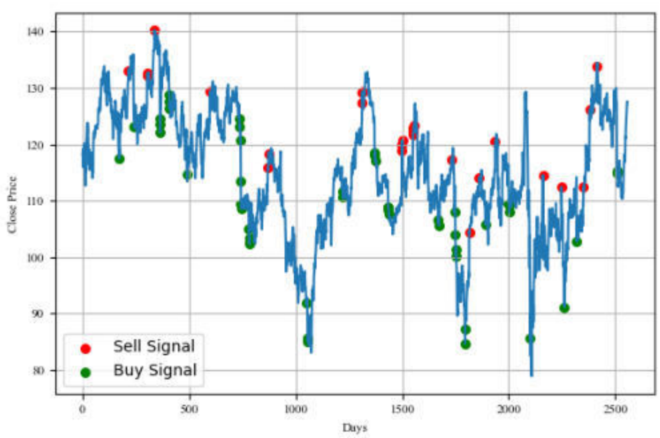


Fig. 2. IBM historical close prices with Bollinger Bands buy and sell signals resulted in a value of $18,208.63 for the test dataset

Comparing the performance of DQN models against benchmark strategies, we observe that each of the DQN configurations consistently outperform RSI, MACD, and Bollinger Bands strategies, showcasing the potential of reinforcement learning approaches in financial decision-making. Among the DQN configurations, those utilizing the Temporal Attention Network denoising method consistently achieved the highest cumulative investment values. This suggests that the attention mechanism employed by TAN effectively captures relevant temporal patterns in the stock price data, leading to superior trading performance.

## V. Conclusion

In conclusion, this study focuses on utilizing DRL techniques for stock trading, specifically focusing on IBM as a case study. By leveraging DRL models within the DQN framework, we have explored various denoising methods to preprocess historical price data, aiming to enhance the robustness and performance of trading strategies.

Our investigation into denoising techniques, including Wavelet Transform, Temporal Attention Network, and Fourier Transform, has revealed promising results. Among these methods, TAN-based denoising consistently outperformed others, exhibiting superior performance in terms of yearly returns, Sharpe ratios, and maximum drawdowns. This underscores the importance of effective preprocessing steps in optimizing trading strategies within the DRL framework. Also, a comparative analysis of different DRL architectures such as MLP and LSTM has been done; however, the difference in results is not significant.

Overall, our findings suggest that DRL-based approaches offer promising avenues for enhancing trading strategies in financial markets, particularly when combined with effective preprocessing techniques and neural network architectures. The superior performance of TAN-based denoising methods, along with MLP or LSTM architectures, highlights the potential for developing more robust and adaptive trading systems capable of navigating the complexities of stock market dynamics.

For future research, there are several interesting directions to explore based on the findings and limitations highlighted in this study. Firstly, given the success of our DQN framework in single stock trading, further investigation could be directed towards examining other reinforcement learning algorithms such as Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), or Policy Gradient (PG) algorithms to compare their performance. Moreover, incorporating additional information sources such as textual data from news headlines and investor sentiments could provide valuable insights for improving trading strategies. By leveraging knowledge graphs and sentiment analysis techniques, we can enhance the decision-making capabilities of reinforcement learning agents and achieve better performance.

## References


[1] N. Kanwar, Deep reinforcement learning-based portfolio management. The University of Texas at Arlington, 2019.

[2] W. Wen, Y. Yuan, and J. Yang, "Reinforcement learning for options trading," Applied Sciences, vol. 11, no. 23, p. 11208, 2021, MDPI.

[3] Y. Li, W. Zheng, and Z. Zheng, "Deep robust reinforcement learning for practical algorithmic trading," IEEE Access, vol. 7, pp. 108014–108022, 2019, IEEE.

[4] Y. Li, P. Liu, Z. Wang, and others, "Stock trading strategies based on deep reinforcement learning," Scientific Programming, vol. 2022, 2022, Hindawi.

[5] J. Lee, H. Koh, and H. J. Choe, "Learning to trade in financial time series using high-frequency through wavelet transformation and deep reinforcement learning," Applied Intelligence, vol. 51, pp. 6202–6223, 2021, Springer.

[6] Z. Liang, H. Chen, J. Zhu, K. Jiang, and Y. Li, "Adversarial deep reinforcement learning in portfolio management," arXiv preprint arXiv:1808.09940, 2018.

[7] Z. Zhang, S. Zohren, and S. Roberts, "Deep reinforcement learning for trading," The Journal of Financial Data Science, 2020, Institutional Investor Journals Umbrella.

[8] Z. Xiong, X.-Y. Liu, S. Zhong, H. Yang, and W. Anwar, "Practical deep reinforcement learning approach for stock trading," arXiv preprint arXiv:1811.07522, 2018.

[9] H. Yang, X.-Y. Liu, S. Zhong, and W. Anwar, "Deep reinforcement learning for automated stock trading: An ensemble strategy," in Proceedings of the first ACM international conference on AI in finance, 2020, pp. 1–8.

[10] Y. Li, W. Zheng, and Z. Zheng, ”Deep robust reinforcement learning for practical algorithmic trading,” IEEE Access, vol. 7, pp. 108014–108022, 2019, IEEE.

[11] A. Shavandi, and M. Khedmati, ”A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets,” Expert Systems with Applications, vol. 208, pp. 118124, 2022, Elsevier.

[12] J. Lee, H. Koh, and H. J. Choe, ”Learning to trade in financial time series using high-frequency through wavelet transformation and deep reinforcement learning,” Applied Intelligence, vol. 51, pp. 6202–6223, 2021, Springer.

[13] Q. Tang, T. Fan, R. Shi, J. Huang, and Y. Ma, ”Prediction of financial time series using lstm and data denoising methods,” arXiv preprint arXiv:2103.03505, 2021.

[14] D. Song, A. M. C. Baek, and N. Kim, ”Forecasting stock market indices using padding-based fourier transform denoising and time series deep learning models,” IEEE Access, vol. 9, pp. 83786–83796, 2021, IEEE.

[15] J. Qiu, B. Wang, and C. Zhou, ”Forecasting stock prices with long-short term memory neural network based on attention mechanism,” PloS one, vol. 15, no. 1, p. e0227222, 2020, Public Library of Science San Francisco, CA USA.

[16] A. Halidou et al., ”Review of wavelet denoising algorithms,” Multimedia Tools and Applications, pp. 1–31, 2023, Springer.

[17] J. Lee, H. Koh, and H. J. Choe, ”Learning to trade in financial time series using high-frequency through wavelet transformation and deep reinforcement learning,” Applied Intelligence, vol. 51, pp. 6202–6223, 2021, Springer.

[18] D. Song, A. M. C. Baek, and N. Kim, ”Forecasting stock market indices using padding-based fourier transform denoising and time series deep learning models,” IEEE Access, vol. 9, pp. 83786–83796, 2021, IEEE.

[19] X. Zhang and S. Jiang, ”Application of Fourier transform and Butterworth filter in signal denoising,” in 2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP), pp. 1277–1281, 2021, IEEE.

[20] S. Shih, F. Sun, and H. Lee, ”Temporal pattern attention for multivariate time series forecasting,” Machine Learning, vol. 108, pp. 1421–1441, 2019, Springer.

[21] C. Fan, Y. Zhang, Y. Pan, X. Li, C. Zhang, R. Yuan, D. Wu, W. Wang, J. Pei, and H. Huang, ”Multi-horizon time series forecasting with temporal attention learning,” in Proceedings of the 25th ACM SIGKDD International conference on knowledge discovery & data mining, pp. 2527–2535, 2019.