Gold Price Prediction and Trading Decision Using Machine Learning and Deep Reinforcement Learning

Ahmad Tajuddin Fauzan1, a) and Suraya Nurain Kalid1, 2, b)

*1Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, 63100, Cyberjaya, Malaysia.*

*2Centre for Advanced Analytics, CoE for Artificial Intelligence, Multimedia University, Persiaran Multimedia, 63100, Cyberjaya, Malaysia.*

b) [corresponding author: suraya.nurain@mmu.edu.my](mailto:%20corresponding%20author:%20suraya.nurain@mmu.edu.my)

a) ahmadtajuddinfauzan@gmail.com

**Abstract.** This study aims to develop and evaluate deep reinforcement learning models for trading in the gold market. Forecasting gold prices is a complex task influenced by numerous economic and market factors. While previous studies have demonstrated the effectiveness of machine learning in predicting gold prices, they have not provided actionable strategies for investors. This research addresses this gap by exploring deep reinforcement learning models to optimize trading strategies. Historical gold data spanning a decade is used to train and evaluate the models. Deep learning models Transformer—are combined with reinforcement learning models—Proximal Policy Optimization (PPO) to create deep reinforcement learning models. The models' performance will be evaluated using Cumulative Return, Sharpe Ratio, and Maximum Drawdown. The findings will contribute to the optimization of stock trading strategies through deep reinforcement learning and offer a structured approach to addressing financial market complexities.

# IntroDUCTION

Gold has been a historical asset and a safe haven in times of economic and geopolitical instability. The interaction of supply and demand is the underlying factor that determines its price and the factors that affect this interaction are not limited to financial markets alone but other factors like industrial use in technology, jewelry use and central bank buying. There are numerous modern and traditional ways that investors can participate in gold trading, such as physical gold bullion, very liquid gold exchange-traded funds, and more speculative means such as futures and options contracts. This variety in investment methods and avenues, ranging between long-term holding to short-term day trading, enables the investors to put their strategy accordingly, in line with their own financial objectives. Nevertheless, the gold market is complex and volatile which presents a challenge. The price of gold is notoriously hard to forecast due to a non-linear interplay of variables such as inflation, interest rates and the value of the U.S. dollar, which is the currency in which gold is usually priced. An appreciation in the dollar, such as the case, may render gold to be more costly to foreigners, and this may lead to a decline in demand and consequent price decline. Sudden international developments, such as pandemics, geopolitical tensions, and other events, can also cause sharp and unpredictable price changes, frequently producing a knock-on effect in several financial markets. That being the case, conventional statistical models and predictive algorithms are not able to keep up with the changing market dynamics. This has caused an increased interest in more complex AI methods, especially Deep Reinforcement Learning (DRL). The difference between DRL and other price predictors is that, DRL does not only predict prices, but rather learns to make the best, sequential trading decisions in real-time. A DRL agent can learn to maximize returns by trading in the market environment and getting feedback in the form of rewards or penalties. The method enables the system to adjust to the complicated, dynamic character of the gold market better than traditional practices. Although promising, there are a number of obstacles that challenge the use of DRL in gold trading. The key issues are ensuring the quality of data, creating generalizable and robust models that can be applied in different market conditions, and increasing the interpretability of models to create trust. These problems need to be addressed to come up with dependable and efficient AI-powered trading systems capable of overcoming the complexity of the gold market.

# RELAted work

The challenges overcome by traditional Recurrent Neural Network such as the problem with vanishing gradients led to the development of Long Short Term Memory networks first introduced by Hochreiter and Schmidhuber in 1997. To do so, the model has a gating mechanism that determines the flow of information in a memory cell, ensuring that important data is kept over long sequences [1]. In Dhuhita et al. [2], LSTM was applied for gold price prediction on historical data from Yahoo Finance. They used hyperparameter tuning with grid search to tune the architecture of the model, a big decrease in Root mean square (RMSE) deviation (0.00033). Other than that, Transformer-based models offer an alternative by capturing intricate relationships within historical price data through self-attention mechanisms. Kumar and Rizk [3] proposed a Transformer-based Reinforcement Learning Model for Optimized Quantitative Trading, which integrates a Transformer encoder-decoder network for stock price prediction with a reinforcement learning agent that optimizes trading strategies. Further advancements in multi-Transformer networks have been proposed to improve Deep Reinforcement Learning-based trading strategies. Next, for deep learning methods, particularly reinforcement learning, have revolutionized decision-making in financial markets by enabling autonomous systems to decide whether to buy, hold, or sell assets [4]. Ekarittikrai and Rattagagan [5] proposed a market making strategy enhancement using PPO and A2C algorithms with Avellaneda-Stoikov model. Suliman et al. [6] used the Dueling Deep Q-Network (DDQN) for cryptocurrency trading. Zhao et al. [7] applied DRL on the real estate with the novel hybrid model via feature extraction combining LSTM and the Gramian Angular Field (GAF). This approach allowed the market data to be captured in terms of temporal and spatial relationships, which subsequently led to more profitable trading strategies. It demonstrated DRL’s ability to solve buy, hold and sell decisions in low frequency trading environments.

# Theoretical frameworks

## Deep Learning Model

### Transformer

The Transformer model is a deep learning architecture designed to handle sequential data efficiently by leveraging a mechanism called self-attention. The Transformer has revolutionized natural language processing (NLP) and time-series forecasting by enabling parallel computation and capturing long-range dependencies more effectively than traditional recurrent architectures like LSTMs and Gated Recurrent Unit (GRU). Unlike Recurrent Neural Networks (RNNs), which process data sequentially, Transformers allow simultaneous processing of all time steps, making them significantly faster and more scalable for large datasets. There are two main components in the Transformer architecture, namely encoder and decoder, which are each composed of multiple layers of self-attention mechanisms and feed forward neural networks [2].

## Reinforcement Learning Model

### Proximal Policy Optimization (PPO)

The goal of Policy Gradient Methods is to directly optimize the policy, i.e. by maximizing the expected reward. Nevertheless, they tend to be unstable and inefficiently learn because of either large or uncontrolled policy updates. Proximal Policy Optimization (PPO) solves these problems by adding a clipped objective function that constrains the amount of policy change during each update. This enhances stability in learning and eliminates radical policy change, which may result in a deterioration of performance. PPO is best suited to financial market forecasting since it can handle continuous decision making (e.g. adjustment of investment positions) and live with volatile markets. It ensures stable learning even in those environments where the uncertainty is high, and it is quite appropriate in financial forecasting and optimization of trading strategies due to its stability [8].

## Technical Indicators

Two basic methods of smoothing financial time series and emphasising longer-term trends are simple moving averages (SMA) and exponential moving averages (EMA). The SMA of period n is merely the arithmetic mean of the last n data points, which smooths daily volatility by assigning the same weight to each observation [9]. In comparison, the EMA exponentially down-weights the older observations, and thus is more sensitive to price movements in the recent past [5]. The moving average convergence divergence indicator builds on EMAs by calculating the difference between two EMAs, 12-day minus 26-day, to measure the strength and direction of a trend, and a 9-day EMA of the difference is used as a so-called signal line whose crossings are used to buy or sell signals.

## Decision-Making Framework (Buy, Hold, Sell)

The state inputs generated by the deep learning model underlying the decision-making framework of a deep reinforcement learning (DRL) trading agent are used to select between three fundamental actions of buy, sell or hold based on whether the model predicts an upwards movement, a downwards movement or the neutral/uncertain state of the market. When the agent decides to purchase, it will either get into or add to its position in the expectation of an upward movement in prices; when it decides to sell, it will liquidate or decrease its position in the expectation of a downward movement; and when it sits, it will continue with the current position in the absence of a clear indication. After each action, the agent receives a reward signal positive for profitable trades and negative for losses or suboptimal decisions—which it uses to update its policy via reinforcement learning. Over time, this reward‐driven feedback loop enables the agent to refine its strategy, gradually improving its ability to capitalize on market movements.

# METHODOLOGY

Figure 1 illustrates the research methodology for this project. The research begins with a defined Research Methodology, followed by Data Collection to prepare the information. Feature Engineering creates new variables, and Exploratory Data Analysis helps understand the data before Feature Selection identifies key inputs. For time-based data, a Sequential Train-Test Split is used. Deep Learning for Price Prediction builds forecasting models, which then inform DRL for Trading Decisions, where an agent learns to trade. Optuna hyperparameter fine-tunes the trading agent, and finally, Result Analysis evaluates the outcomes, concluding the research.

A diagram of a software development process

AI-generated content may be incorrect.

**FIGURE 1.** Research overview

## Data Collection

The dataset used in this study is a dataset that contains historical gold prices. The dataset was spanned from 28th January 2015 to 1st January 2025 (10 years) to ensure the size of dataset is large enough to be analysed. The gold price is obtained from public site and consists of the date, open, high, low, close, adjusted close, and volume. An initial validation of the data was performed on collection, which involved verifying the integrity and accuracy of the data, to ensure that the data accurately reflects the gold price chosen. In this research project, the chosen integrated development environment (IDE) is Jupyter Notebook. Jupyter Notebook is an open-source web application that is widely used for data science and machine learning project.

## Data Preprocessing

Preprocessing of data is a basic component of data analysis and modeling process. It is the process of organizing raw, unstructured and incomplete data into a clean and structured format which is then fit to be analyses. This is done to deal with the usual data problems of missing values, inconsistencies, noise, and irrelevant data to ensure that the resulting dataset is accurate and reliable prior to any additional analytical efforts.

The initial step of preprocessing is the detection and elimination of duplicates and treatment of missing values. With the help of Python functions such as isnull() and duplicated(), it is possible to scan the dataset quickly and identify null values and duplicates in every column and row. This is necessary to ensure quality and integrity of the data to be used in training and testing predictive models. Data transformation comes next and this is where a transformation of data into consistent and usable formats takes place. As an example, the dates can be re-arranged to eliminate the leading zeros, and the volume numbers that are written in the form of K (to represent thousands) are being converted into the full numeric values by multiplying by 1,000. These changes make all fields in the right format to perform calculations and subsequent analysis.

Lastly, feature engineering is carried out to augment the dataset by generating new features that can assist the models identify the underlying patterns. Technical indicators Simple Moving Average (SMA), Exponential Moving Average (EMA), and Moving Average Convergence Divergence (MACD) are inserted as new columns in this study. Such indicators can be used to give a good idea about the trend and momentum of the market, and both machine learning and deep reinforcement learning models can use them to make more informed predictions and trading decisions. Any missing values that do occur as a result of these calculations, e.g. as a result of rolling window calculations, are dealt with in an appropriate manner so that the data remains consistent throughout the dataset.

## Deep Reinforcement Learning for Trading

In this project, the model of DRL is PPO. Table 1 shows the trading environment was set where the gold market simulation for the model to interact with the DRL model development. The environment platform where the agent is able to learn when to buy, hold or sell choices through the current market state. We introduce the MDP framework which structures the trading environment allowing agent to navigate the complexity through decision making under uncertainty. The agent learns to maximize rewards while it learns to adapt to changing market conditions across this framework. The rest of the sections define the core (trading) environment components that may allow the agent to learn and perform effective trading actions.

|  |  |
| --- | --- |
| **TABLE 1.** Components of trading environments | |
| **Components** | **Description** |
| State Space (S) | Open, High, Low, SMA, EMA, MACD, Signal\_Line, Close |
| Action Space (A) | Sell (0), Hold (1), Buy (2) |
| Reward Function (R) | Closing Price (today) – Closing Price (yesterday) |

The financial features include the current price, daily high, daily low, SMA, EMA, and MACD in its state space. To select these features, RFE is used and the features are important to make trading decisions with information. The three discrete actions we consider as the action space are buy, hold, sell. The reward function is positive. reward for profitable trades and a negative reward for losses, and it is based on the profitability of the agent’s actions.

## Performance Metrics

In this study we apply the Root Mean Square Error (RMSE) and the coefficient of determination (R2) to test the performance of our prediction models. RMSE measures the average error of the predictions and the lower the value the better the forecast. R2 is a measure of the goodness of fit of the model predictions to the variance of the actual values the closer it is to 1 the better the fit. In the case of Deep Reinforcement Learning (DRL) trading strategy, we evaluate the performance by Sharpe Ratio and Maximum Drawdown. It is a popular risk-adjusted measure of performance that is calculated as the ratio of the mean excess return (over a risk-free benchmark) to the standard deviation of returns, the Sharpe Ratio. This measure gives an indication of the effectiveness of the strategy in translating risk to reward, the higher the Sharpe Ratio the better the performance relative to risk taken. It enables investors to compare various strategies not only on the basis of their returns, but also on the basis of the amount of volatility that was necessary to obtain the returns. The second measure is Maximum Drawdown which is the maximum decline noted between the highest value of the portfolio and the lowest value over the period of evaluation. It is the measure of the worst loss scenario that is particularly important in measuring capital preservation and downside risk. The smaller the maximum drawdown, the more stable and robust the strategy is, even under unfavorable market conditions.

# Results AND Discussion

## Deep Learning Price Prediction

Table 2 compares predicting accuracy for four deep-learning models on gold-price data using Open, High and Low as inputs. The Simple RNN leads the pack with an RMSE of 0.01104 and R² of 0.99722, closely followed by the LSTM (RMSE: 0.01338, R²: 0.99592). The Transformer sits in third place (RMSE: 0.01797, R²: 0.99263), achieving respectable performance given its smaller embedding dimension and limited hyperparameter tuning. Next, CNN records the highest error (RMSE: 0.01901, R²: 0.99176). Although our LSTM and RNN models showed slightly better RMSE and R² in the supervised forecasting tests, we chose the Transformer as the state encoder for our PPO-based reinforcement-learning system for three main reasons. First, its self-attention layers excel at capturing long-range temporal patterns in price data. Second, recent studies demonstrate that Transformers handle delayed rewards and complex state representations particularly well in RL settings. Third, their highly parallel architecture scales efficiently to larger and richer financial datasets. These combined strengths make the Transformer a natural fit for algorithmic‐trading applications.

|  |  |  |
| --- | --- | --- |
| **TABLE 2.** Performance comparison of deep learning models for gold price prediction | | |
| **Model** | **RMSE** | **R2** |
| LSTM | 0.013380 | 0.995917 |
| RNN | 0.011044 | 0.997218 |
| CNN | 0.019005 | 0.991762 |
| Transformer | 0.017970 | 0.992630 |

## DRL Model Evaluation Result

Table 3 summarizes the performance metrics of a Deep Reinforcement Learning (DRL) model across four datasets: Gold, Nvidia, Oracle, and Microsoft. The results show that the DRL model achieves its best risk-adjusted performance on the Gold dataset, where an initial capital of $100 grows to $187.17, resulting in total profits of $87.15, a high Sharpe Ratio of 3.60, and a very low maximum drawdown (MDD) of 3.39%. In comparison, Nvidia provides the highest total profits ($253.49) and final capital ($353.49), with a Sharpe Ratio of 1.63 and an MDD of 26.45%. Oracle achieves a final capital of $182.64 (profit of $82.64), a Sharpe Ratio of 1.07, and an MDD of 36.17%. Microsoft shows the lowest growth, with a final capital of $121.54 (profit of $21.54), a Sharpe Ratio of 0.45, and an MDD of 32.07%. The overall average Sharpe Ratio across all datasets is 1.69, with a cumulative profit of $111.21 and an MDD of 24.52%. Excluding gold, the average Sharpe Ratio drops to 1.05, with average profits of $119.22 and a higher MDD of 31.56%. These results highlight the model’s strong risk-adjusted returns in gold, while its performance on stocks is profitable but riskier.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TABLE 3.** Performance metrics results for DRL | | | | | |
| **Dataset** | **Initial Capital** | **Final Capital** | **Total Profits** | **Sharpe Ratio** | **Maximum Drawdown (%)** |
| Gold | 100 | 187.17 | 87.15 | 3.60 | 3.39 |
| Nvidia | 100 | 353.49 | 253.49 | 1.63 | 26.45 |
| Oracle | 100 | 182.64 | 82.64 | 1.07 | 36.17 |
| Microsoft | 100 | 121.54 | 21.54 | 0.45 | 32.07 |
| **Overall Average** | | | **111.21** | **1.69** | **24.52** |
| **Average without Gold Commodities** | | | **119.22** | **1.05** | **31.56** |

## Comparison with Existing Works

Table 4 clearly shows that the current PPO-Transformer model delivers strong and competitive overall performance compared to previous methods. With a Sharpe Ratio of 1.05, the PPO-Transformer outperforms LSTM-DDPG (0.74) and DQN-GRU (0.69), indicating improved risk-adjusted returns over most prior models. However, it is slightly lower than DADE-DQN (1.20) and TDQN (1.48), which remains the highest among the models compared. In terms of Maximum Drawdown (MDD), the PPO-Transformer records -31.56%, which is comparable to LSTM-DDPG (-30.91%) and DQN-GRU (-34.20%), but higher (worse) than DADE-DQN (-13.95%) and TDQN (-17.31%). This suggests that while the PPO-Transformer offers strong risk-adjusted returns, it does so with a higher exposure to drawdown risk compared to the best-performing past models. Overall, the PPO-Transformer stands out as a robust and effective model in terms of risk-adjusted returns, offering competitive performance among recent state-of-the-art approaches.

|  |  |  |  |
| --- | --- | --- | --- |
| **TABLE 4.** Result comparison with past works | | | |
| **Author** | **Model Used** | **SR** | **MDD (%)** |
| Huang et al. [8] | DADE-DQN | 1.20 | - 13.95 |
| Fu et al. [10] | LSTM-DDPG | 0.74 | - 30.91 |
| Ansari et al. [11] | DQN-GRU | 0.69 | - 34.20 |
| Théate and Ernst [12] | TDQN | 1.48 | - 17.31 |
| **Proposed Model** | **PPO-Transformer** | **1.05** | **- 31.56** |

# Conclusion

This research presents a deep reinforcement learning framework, the PPO-Transformer, which integrates Proximal Policy Optimization (PPO) with a Transformer-based architecture to enhance temporal modelling and decision-making capabilities. Unlike traditional models such as Linear Regression or Random Forest, the Transformer component effectively captures long-term dependencies in sequential financial data, leading to more informed and stable policy learning. The PPO-Transformer was evaluated across four financial datasets which is Gold, Microsoft, Oracle and Nvidia where it achieved a strong average total profit of 111.21, demonstrating superior annualized returns. Moreover, the model achieved a Sharpe Ratio (SR) of 1.05 and a Maximum Drawdown (MDD) of -31.56%, reflecting a balanced trade-off between return and risk. Comparative analysis with other state-of-the-art DRL models, including DADE-DQN, LSTM-DDPG, DQN-GRU, and DQN-LSTM, shows that the PPO-Transformer is competitive in terms of performance and robustness. Additionally, the modular nature of the proposed framework offers strong potential for generalization to other DRL tasks beyond financial domains.

# References

1. N.P.J.R. Dewi, N.L.W.S.R. Ginantra, I.W.A.S. Darma, and I.G.A. Indrawan, “Application of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) Algorithm in Gold Price Prediction,” in *2023 11th International Conference on Cyber and IT Service Management (CITSM)*, (IEEE, Makassar, Indonesia, 2023), pp. 1–6.
2. W.M.P. Dhuhita, M.F.A. Farid, A. Yaqin, H. Haryoko, and A.A. Huda, “Gold Price Prediction Based On Yahoo Finance Data Using Lstm Algorithm,” in *2023 International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS)*, (IEEE, Jakarta Selatan, Indonesia, 2023), pp. 420–425.
3. A. Kumar, R. Rizk, and K. Santosh, “Transformer-based Reinforcement Learning Model for Optimized Quantitative Trading,” in *2024 IEEE Conference on Artificial Intelligence (CAI)*, (IEEE, Singapore, Singapore, 2024), pp. 1454–1455. U. Suliman, T.L. Van Zyl, and A. Paskaramoorthy, “Cryptocurrency Trading Agent Using Deep Reinforcement Learning,” in *2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, (IEEE, Toronto, ON, Canada, 2022), pp. 6–10.
4. T. Kabbani, and E. Duman, “Deep Reinforcement Learning Approach for Trading Automation in the Stock Market,” IEEE Access **10**, 93564–93574 (2022).
5. C. Ekarittikrai, and E. Rattagan, “Enhancing Market Making Strategies with Deep Reinforcement Learning-Based Quoting Decisions,” in *2024 21st International Joint Conference on Computer Science and Software Engineering (JCSSE)*, (IEEE, Phuket, Thailand, 2024), pp. 324–329.
6. U. Suliman, T.L. Van Zyl, and A. Paskaramoorthy, “Cryptocurrency Trading Agent Using Deep Reinforcement Learning,” in *2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, (IEEE, Toronto, ON, Canada, 2022), pp. 6–10.
7. Y. Zhao, G. Chetty, and D. Tran, “Deep Learning for Real Estate Trading,” in *2022 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, (IEEE, Gold Coast, Australia, 2022), pp. 1–7.
8. Y. Huang, X. Lu, C. Zhou, and Y. Song, “DADE-DQN: Dual Action and Dual Environment Deep Q-Network for Enhancing Stock Trading Strategy,” Mathematics 11(17), 3626 (2023).
9. A.A.S. Gunawan, S. Bilqis Ashifa, R.Y. Rumagit, and H. Ngarianto, “Development of Stock Market Price Application to Predict Purchase and Sales Decisions Using Proximal Policy Optimization Method,” in *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, (IEEE, Jakarta, Indonesia, 2021), pp. 431–437.
10. K. Fu, Y. Yu, and B. Li, “Multi-Feature Supervised Reinforcement Learning for Stock Trading,” IEEE Access 11, 77840–77855 (2023).
11. Y. Ansari, S. Yasmin, S. Naz, H. Zaffar, Z. Ali, J. Moon, and S. Rho, “A Deep Reinforcement Learning-Based Decision Support System for Automated Stock Market Trading,” IEEE Access 10, 127469–127501 (2022).
12. Théate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. *Expert systems with applications*, *173*, 114632.