Project Report: Stock Prediction Using Reinforcement Learning

Group- 4

Om Thakur
Kumar Satyanshu
Divyansh Gupta
Meghansh Govil
Mayank Sengupta

1. Introduction

Overview of the Project

Forecasting the future performance of a stock is the work of stock prediction. It is based on historical data and various market indicators. Advancements in the field of artificial intelligence(AI) have given us reinforcement learning(RL) as a promising tool to help with financial applications due to its ability to handle dynamic environments.

Motivation

RL is a subfield of deep learning, it aims to maximise profit while also adapting to changes in the environment it is operating in constantly. Due to the frequency and duration of stock price fluctuations, reinforcement learning (RL) is a more suitable predictive tool for the statistical analysis of data than supervised learning.
This project explores RL-based trading algorithms for portfolio management and evaluates their performance across different datasets and markets.

Objectives

- Compare the performance of trading algorithms like Deep Scalper, EIIE, and Qlib.
- Test these algorithms on diverse datasets, including DJ30 (American stocks) and Indian stock markets.
- Explore portfolio management strategies using RL frameworks.

Structure of the Report

The report consists of background research, experimental phases, technical implementation details, challenges encountered, and future directions.

2. Background and Literature Review

Fundamentals of Reinforcement Learning in Finance

In RL, an agent interacts with an environment like financial markets to learn and predict the best action(buy, sell, hold) based on reward.
Reinforcement learning involves an agent interacting with an environment (e.g., financial markets) to learn optimal actions (buy, sell, hold) based on rewards (e.g., profit). Key terms include:

- State: Current market conditions.
- Action: Trading decisions.
- Reward: Feedback based on financial gains or losses.

Review of Trading Algorithms

- Deep Scalper: Designed for intraday cryptocurrency trading with high volatility but underperformed on DJ30 due to its focus on Bitcoin patterns, which are highly volatile.
- EIIE (Ensemble Investment Intelligence Engine): Utilizes portfolio vector memory and stochastic learning for stock trading. It showed promising results on DJ30.
- Qlib: A comprehensive framework by Microsoft for quantitative trading, offering tools for model development and portfolio optimization.

Key Research Contributions

Studies highlight RL's adaptability in financial markets, especially for portfolio management and risk optimisation.

3. Phase One: Experiments with Deep Scalper on DJ30

Deep Scalper: Overview and Limitations

Deep Scalper is a reinforcement learning (RL)-based trading algorithm designed for intraday cryptocurrency markets, such as Bitcoin. To execute frequent trades, it leverages high market volatility and rapid price fluctuation.

Its key features are:-

1. Dueling Q-Network
2. Hindsight Bonus Reward Function
3. Risk-Aware Mechanisms

It is adapted to a fast-paced environment like cryptocurrency trading as its design helps focus on exploiting short-term price movements.

Why It Was Initially Considered

The reason we initially chose deep scalper was due to its success in the crypto markets, highlighting its ability to handle high volatility and make rapid decisions accordingly.

Limitations Encountered on DJ30

1. Lower Volatility: DJ30 stocks exhibit less volatility compared to cryptocurrencies, reducing the effectiveness of scalping strategies.
2. Mismatch in Data Granularity: Deep Scalper is designed for minute-level trading, so it struggles with DJ30's day-level data.
3. Reward Function Mismatch: The hindsight bonus reward function of the Deep Scalper does not align well with DJ30's slower trends.

These limitations undermined the effectiveness of Deep Scalper for the DJ30 dataset, prompting us to switch our algorithm.

4. Phase Two: Transition to EIIE

Given Deep Scalper's limitations, Ensemble Investment Intelligence Engine (EIIE), a reinforcement-based learning algorithm designed specially for stock prediction, was chosen next. Due to it being designed for portfolio management and long-term investment, it is much more suitable for a long dataset like DJ30.

EIIE helps track and optimize portfolio allocation over time, this is done with the help of portfolio vector memory. Its aim is to maximise portfolio value by using a stochastic policy gradient method while also considering market dynamics. Due to this approach, the EIIE is able to adapt to slower-moving trends in stock markets and perform better than the Deep Scalper.

Results from EIIE Implementation

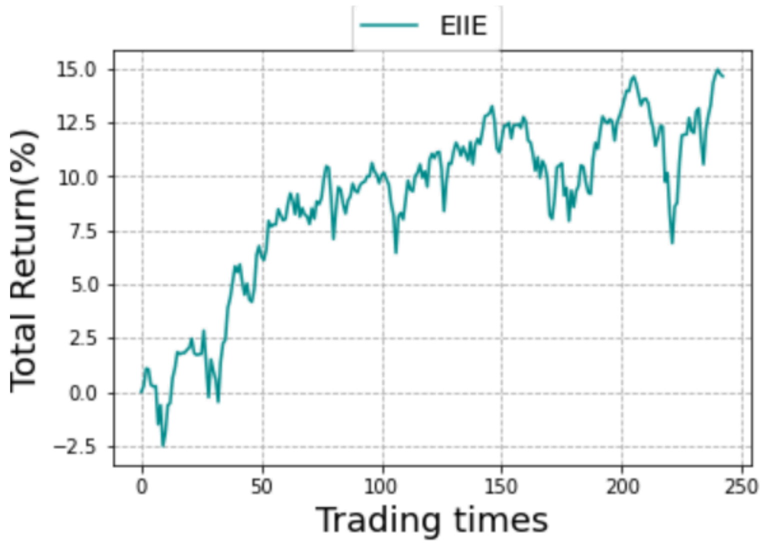

Fig 1: Total return in trading instances

Average drop-down: 8%

- Volatility: 0.7%
- Sharpe ratio: 1.31
- Maximum total return: 14.61%

These metrics indicate that EIIE managed risk effectively while achieving good returns.

Limitations Encountered

However, we aimed to predict stock for newer data and also to predict stock of the Indian market, which was not possible with EIIE. This is mainly due to EIIE's limited ability to generalise across different market conditions and data characteristics. EIIE's architecture is highly effective in optimising portfolio allocation for specific datasets, but it relies heavily on the statistical properties and patterns present in the training data, such as higher volatility in certain sectors, differing liquidity levels, and unique macroeconomic factors. Making EIIE not fit for this job.

Further, EIIE is heavily reliant on historical data, making it difficult to evolve to newer trends and newer datasets. This let us to once again change our RL algorithm.

5. Phase Three: Transition to Qlib

Qlib is an AI-oriented investment platform which was created by Microsoft. The use of artificial intelligence it improves its research and generates value in quantitative investment. Qlib includes various machine learning modeling paradigms, such as supervised learning, market dynamics modeling, and reinforcement learning. It covers the whole machine learning pipeline, from data processing to training of the model, back-testing, and the complete range of quantitative investment processes, including alpha seeking, risk modeling, portfolio optimization, and order execution.

In the context of the "Stock-Platform-Using-EIEE" repository, Qlib is used to tap into its wide-ranging features to develop and deploy machine learning models specific to stock market prediction. Our model aims to develop a platform that shows the world's best stocks with different indicators and types of graphs and applies a reinforcement learning model based on DeepScalper for intraday trading. With the integration of Qlib, our project has a strong infrastructure that simplifies the workflow from data acquisition to model deployment, allowing efficient experimentation and implementation of sophisticated trading strategies.

Qlib is suitable for the dataset and model applied in our project because it supports various machine learning paradigms and has a broad set of tools specifically for quantitative investment. It is designed as a modular tool, which facilitates easy customization and scalability to implement complex models such as DeepScalper. Furthermore, the ability of Qlib to process large-scale data and support back-testing frameworks ensures that training and evaluation can be done with models, resulting in more accurate and sound stock market forecasting.

6. Technical Implementation Details

Data Handling

- Dataset Sources: The dataset was prepared by utilizing Qlib's built-in data provider, which included historical stock market data.
- Stock Selection: The platform targeted global stock indices and the top 10 highest-performing stocks globally, including prominent Indian stocks.
- Preprocessing Techniques:
    1. Utilized rolling window techniques to extract short-term trends.
    2. We used z-score normalization for feature scaling and to make the data standardized.
    3. Feature engineering involved volatility indicators and moving averages to improve model predictions.

Algorithmic Adaptations

- DeepScalper was employed for intraday trading but needed extensive fine-tuning across various market conditions.
- EIIE was optimized using reinforcement learning-based techniques to increase portfolio balancing strategies.

- Qlib's model framework was incorporated, enabling dynamic hyperparameter tuning specific to stock market fluctuations.
- Custom reward functions implemented were aimed at maximizing portfolio returns while reducing risk exposure.

Performance Metrics Explained

- Maximum Drawdown (MDD): Measures worst-case scenario loss from peak to trough.
- Annualized Volatility: Indicates the extent of risk in stock price volatility.
- Sharpe Ratio: Employed to measure risk-adjusted returns for improved validation of strategy.
- Cumulative Return: Indicates the total profitability of the trading strategy.

7. Lessons Learned and Future Directions

Lessons Learned

- Algorithm Specificity**:** The experiments underscored that algorithms like Deep Scalper, while robust in volatile cryptocurrency environments, require significant adaptation for traditional stock markets. The limited data granularity and reward structure mismatches stressed the importance of aligning algorithm design with market-specific characteristics.
- Insights from EIIE Implementation: The repository for the EIIE-based stock platform provided valuable lessons in modular design and code reusability. It demonstrated that a flexible, component-driven architecture can ease the transition between models and streamline parameter tuning for diverse datasets.
- Integration and Adaptation Challenges: Transitioning from crypto-focused strategies to those applicable to indices like DJ30 and emerging markets exposed challenges in risk management and model generalisation. EIIE, though promising, revealed that reliance on historical data and fixed statistical properties can hinder performance when applied to newer market conditions.

Future Directions

- Hybrid Model Development: There is potential to develop hybrid reinforcement learning models that blend the rapid decision-making of Deep Scalper with the robust portfolio optimization capabilities of EIIE. Such integration could address the shortcomings of each approach.

- Real-Time Trading Enhancements: Future work should focus on incorporating real-time intraday trading capabilities, enhanced by adaptive risk management and dynamic hyperparameter tuning, to better capture evolving market trends.
- Extended Market Validation: Expanding experiments to include additional emerging markets and integrating broader data sources—including sentiment analysis and alternative market indicators—could further refine predictive accuracy.
- Scalability and Modular Expansion: Leveraging the modular design showcased in the EIIE repository, future platforms could be scaled using cloud-based environments to test models across larger, more varied datasets, ultimately leading to more resilient and adaptable trading systems.

8. Conclusion

This project experience contains a critical assessment of different reinforcement learning methods we used for stock forecasting. Early tests with Deep Scalper confirmed its usability in high-volatility, crypto-based settings, but its shortcomings in the conventional stock markets proved to outweigh its pros. The EIIE-based model, as seen in the cited repository, was a more flexible method that enhanced our portfolio management with the help of a modular structure with difficulties in using the new data. Finally, the switch to Qlib helped in the management of varied market flows and registered favorable returns on DJ30 and Indian stock datasets.

The comparative study of these models further encourages the strength of reinforcement learning in finance. Although every algorithm is diverse in terms of strengths, the findings and challenges faced promote the creation of hybrid models and real-time adaptive systems. These prospective directions not only guarantee improved performance but also lead to the general advancement of AI-driven trading strategies for a more sophisticated financial world.

Details about Code Base

Simple notebooks and data CSV files are attached in the file; however, due to the large nature of the files, they are extensively documented in the GitHub links below (for both DeepScalper&EIIE and QLib).

Github link for Phase 1&2-https://github.com/SuperPowered-Cat/Stock-Platform-Using-EIIE
Github link for Phase 3-https://github.com/SuperPowered-Cat/Indian-Stock-Manager